AMSAA
Excellence in Analysis

TECHNICAL REPORT NO. TR-2006-4

DEVELOPMENT OF A MATHEMATICA TOOL FOR IMPLEMENTATION OF A PROGNOSTICS DECISION-MAKING PROCESS BASED ON COMPONENT LIFE HISTORY

MARCH 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

U.S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
ABERDEEN PROVING GROUND, MARYLAND 21005-5071

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so specified by other official documentation.

WARNING

Information and data contained in this document are based on the input available at the time of preparation.

TRADE NAMES

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. The report may not be cited for purposes of advertisement.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2006	Technical Report

4. TITLE AND SUBTITLE
Development of a Mathematica Tool for Implementation of a Prognostics Decision-Making Process Based on Component Life History

5. FUNDING NUMBERS

6. AUTHOR(S)
Dr. Michael J. Cushing

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Director
U.S. Army Materiel Systems Analysis Activity
392 Hopkins Road
Aberdeen Proving Ground, MD 21005-5071

8. PERFORMING ORGANIZATION REPORT NUMBER
TR-2006-4

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Director
U.S. Army Materiel Systems Analysis Activity
392 Hopkins Road
Aberdeen Proving Ground, MD 21005-5071

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)
The key benefit of prognostics is that it can be used to reduce failure risks during deployments and missions when failure is particularly disadvantageous and maintenance inconvenient due to the reduced logistics footprint. One approach to prognostics is to monitor usage in conjunction with an aging model thereby keeping track of remaining component lifetime. This enables one to track usage with on-board sensors and embed an algorithm in on-system logistics software that will automatically generate maintenance alerts and recommendations so that a covered component can likely be replaced before failure as its remaining lifetime decreases and failure risk increases. An additional benefit of usage-based prognostics is that it can also be used to identify an optimum replacement age that minimizes life cycle costs for components that age, provided the costs of in-service failure are greater than planned replacement which is often the case. This report documents the development and application of a collection of functions written in *Mathematica* that can be used to implement usage-based prognostics using life distributions for components that become less reliable with usage.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

THIS PAGE INTENTIONALLY LEFT BLANK.

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Usage-Based Prognostics

A key benefit of prognostics is that it can be used to reduce failure risks during deployments and missions when failure is particularly disadvantageous and maintenance inconvenient due to a reduced logistics footprint. One approach to prognostics is to monitor usage in conjunction with an aging model which allows one to track remaining component lifetime throughout the lifecycle in on-system maintenance software. This approach is termed usage-based prognostics and is depicted in Figure 1-1.

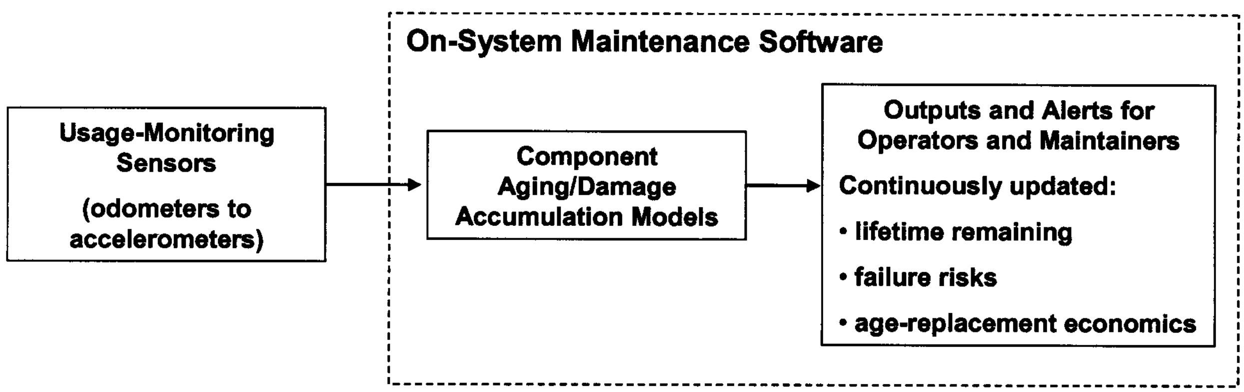

Figure 1-1 Usage-Based Prognostics

This prognostics approach automatically generates maintenance warnings/recommendations that enable one to schedule the replacement of a component as its remaining lifetime decreases, its failure risk increases, and/or it is most economical to do so. An example of usage-based prognostics is the algorithm found in many new cars that is used to track the remaining oil life based on usage history tracked in terms of engine revolutions and temperature. One approach to usage-based prognostics is keep the tracking of usage relatively simple, typically by tracking operating time, distance traveled, revolutions, cycles and/or rounds fired. This was termed life history-based prognostics in AMSAA Technical Report 736 (TR-736) and one should refer to that report for additional discussion of the approach.

This report is AMSAA's second on the topic of life history-based prognostics. TR-736 was the first report and describes usage-based and life history-based prognostics in greater detail. TR-736 documents functions for calculating various reliability and failure-risk metrics in advance of a deployment or mission. One may wish to use these functions in on-system logistics software to recommend when to replace aging components before they fail, as part of a usage-based approach to prognostics that focuses exclusively on management and mitigation of failure risks. Mitigating deployment/mission failure risks has economic consequences in that replacing a component before failure results in the loss of its remaining lifetime, a topic that was only briefly addressed in TR-736. It may be prudent to give up a quantity of lifetime in order to avoid the

additional costs of in-service failure. This new report addresses these economic considerations so that age-replacement economics can be considered along with mitigation of deployment/mission risks.

The primary focus of the previous report was to adapt life history-based prognostics to the cyclic nature of military deployments and missions. The primary focus of this report is on folding on-system, automated age-replacement policies into that mix. Age replacement means that a particular component on a system is replaced either when it reaches a specific age or fails in-service, whichever occurs first. Age-replacement policies are appropriate for components that meet two conditions:

1. The component ages (i.e., it has a failure rate or hazard function that increases with age or usage).
2. Failure of a component during actual operation (e.g., failure during a military deployment or mission) is more costly (i.e., in terms of financial cost or downtime) than replacing it at a pre-determined age under preferred circumstances.

An additional benefit of including on-system age-replacement economics in a usage-based prognostics approach is that it can also be used to identify an optimal replacement age that minimizes life cycle costs for components that age, provided the costs of in-service failure are greater than those of planned replacement. Sources of additional costs due to in-service failures include:

- lost system usage (e.g., fuel, dollars per flight hour, wear & tear) due to an aborted mission,
- lost operator and crew time until system is repaired,
- high-priority shipping of parts,
- recovery personnel time, and
- fuel for, and wear & tear on, recovery vehicles.

Structure of This Report

Chapter 2 provides a conceptual overview of how one might implement usage-based prognostics in on-system maintenance software using the new functions herein, in combination with those in TR-736. The conceptual overview requires the use of functions for performing two categories of computations. The first category is computation of deployment/mission reliability and the second is computation of age-replacement economics. TR-736 provides many of the functions for the first category whereas this report provides the rest.

Chapter 3 presents and illustrates the use of a collection of new functions for on-system computation of age-replacement policies. The use of many of the key functions is illustrated with the track component from TR-736. Processes that use these functions for obtaining age-replacement policies that are optimum with respect to either financial cost or availability are also presented. The functions and processes permit one to consider the long-term economic or availability consequences of establishing component age-replacement policies. The method considers both the loss of remaining lifetime and the consequences of in-service failure in order to arrive at an optimal solution. Use of the functions developed in TR-736, in conjunction with the functions developed herein, will allow one to consider both the mission/deployment risks as well as the long-term benefits of replacing an aging component before it fails.

Chapter 4 presents and applies functions for performing age-replacement simulations. These simulations are used to obtain optimal replacement ages when the currently-installed component is not necessarily new and the time horizon is finite, as is typically the case before a deployment or mission. The functions and process illustrated in Chapter 4 would be useful in cases where one is developing maintenance-planning software that is to be run before a deployment, mission, exercise, or some other life-cycle segment where the cost of an in-service failure is considerably higher than the cost of an age replacement. Using the track component from TR-736, three illustrative cases are examined in detail in Chapter 4.

Chapter 5 summarizes the report and makes recommendations based on the key results of Chapters 3 and 4. Two areas that should be considered for follow-on efforts are identified and briefly described.

In addition, this technical report documents the development of new tool (in the form of an add-on *Mathematica* package) that implements the functions used herein. The effort was undertaken because transparent reliability software which implements these functions is not readily available. This tool is an extension of *Mathematica* and the new functions can be found in Appendix A. Installation instructions for the package are provided as Appendix B and Appendix C documents how the new functions were checked. An updated version of the package *Reliability`ConditionalDistributions`*, which was originally developed with TR-736, is provided as Appendix D.

The electronic form of each chapter and appendix of this report is a *Mathematica* 5 notebook. All of the methodology, computations and graphics in this report are *Mathematica* executables. The results were generated and inserted by *Mathematica*. Thus the technical content of this report is "live" in the sense that it can be re-executed as desired by readers working with the electronic version (provided they have a copy of *Mathematica*). Please refer to *The Mathematica Book* [Wolfram 2003] for information on this software. Additional information, including a free *Mathematica* reader, is available at *http://www.wolfram.com/*.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

On-System Implementation of Life History-Based Prognostics: An Initial Conceptual Overview

Introduction

Our first technical report on life history-based prognostics was AMSAA Technical Report 736 (TR-736). That report focused primarily on deployment and mission risks due to components that become less reliable with usage or age. The figures and tables in that report provided valuable insight concerning when one might want to replace an aging component in advance of a deployment or mission. One topic not addressed in TR-736 was the economics of component age replacement. This technical report incorporates such considerations and integrates them with the earlier work.

Another topic not addressed in TR-736 was how one might implement life history-based prognostics on-board a system. This chapter provides an initial conceptual view of just such an implementation. The implementation concept discussed herein is a first step. It is anticipated that more definition will be required in order to address practical issues that may arise.

On-System Implementation

The application of life history-based prognostics should begin during the system-development phase and continue throughout the life cycle. Efforts that must be performed during development include identifying and selecting components that age, determining applicable lifetime distributions and parameters, and considering how best to implement the process on-board the system. Figure 2-1 depicts a conceptual approach to the on-system implementation of prognostics that is based on life history. The key blocks in the figure will be discussed at length in the subsections that follow it.

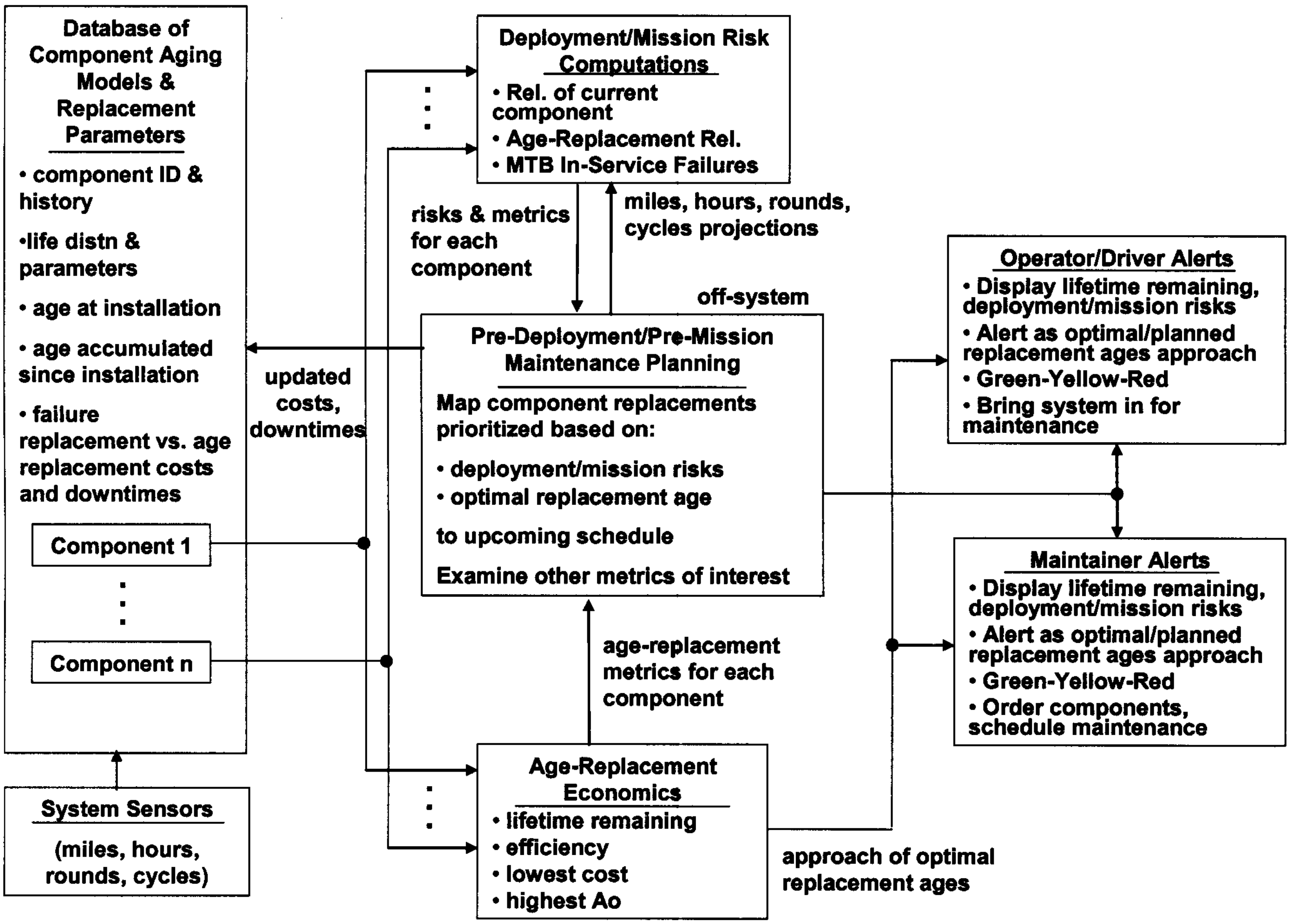

Figure 2-1 On-System Implementation of Life History-Based Prognostics

▪ On-System Database

There should be an on-system database that contains component aging models and replacement parameters. The database should contain information on each component, to include:

- component identification,
- component history,
- the appropriate life distribution and its parameters,
- component age (i.e., distance, time, rounds and/or cycles, as appropriate) at the time it was installed in the system,
- the distance, time, rounds and/or cycles accumulated since the component was installed,
- the cost of a failure-triggered replacement,
- the cost of a planned age replacement,
- the system downtime expected to result from a failure, and
- the system downtime expected to result from a planned age replacement.

When a component is first installed on a system, data such as its national stock number, manufacturer identification, and its repair history, if any, should be known in order that the maintainer can select the most appropriate model and parameters from the database. Its current age at time of installation (i.e., in terms of miles, hours, cycles or rounds accumulated while on another system) should also be entered so that the component's age can be tracked from that point forward.

As the system is operated, system sensors will provide data on the usage that the component experiences and the database should automatically be updated to reflect this usage. The measures of usage can be in terms of distance, time, rounds and/or cycles, as well as a combination of these. The appropriate usage measures will depend upon what drives the reliability of each component.

It is expected that the on-system component database will also need to be updated on a periodic basis:

- Updated parameter estimates (e.g., Weibull distribution parameters) may be received for components that are already in the on-system database. The updated parameter estimates may result, for example, from the analysis of additional or more-recent data.
- New models for components not already in the on-system database will periodically be received. Instances of this may be due to a new source for a component. Components that are repaired or overhauled may have different parameters than when they were new.

The manner in which on-system databases will be updated is an important implementation issue that will need to be addressed during system development.

The database will feed component data to:

- mission-oriented algorithms so that reliability calculations can be made for upcoming deployments and missions, and
- economic-oriented algorithms so that calculations of optimal replacement ages can be made.

■ Deployment and Mission-Risk Computations

Components that are subject to aging become more likely to fail as they accumulate usage. In advance of a deployment or mission, one can calculate the probability that selected components will fail during the deployment or mission. If failure risks are sufficiently high, one may want to replace one or more system components in order to reduce the likelihood of failures occurring under less-than-ideal circumstances. Key metrics that one may want to compute in order to assess component failure risks during a deployment or mission are:

- The probability that a component currently installed will fail (or not fail) as a function of operating miles/hours/cycles/rounds given the current age of the component.
- The probability that a component currently installed will fail (or not fail) in the next x miles/hours/cycles/rounds given the current age of the component.
- The probability that a succession of components that are managed under an age-replacement policy will not suffer an in-service failure.
- The expected distance, time, cycles or rounds between in-service failures given a specific age-replacement policy for the component.

Mathematica functions for performing calculations, such as the first two in the list above, can be found in TR-736. The implementation of functions for the age-replacement calculations (i.e., the third and fourth bullets above) can be found in

subsequent chapters of this report. It is intended that algorithms that implement these *Mathematica* functions will be embedded in software that resides on-board the system.

As depicted in Figure 2-1, performing these deployment and mission risk computations requires data from the component databases as well as projections of the quantities of distance, time, cycles and rounds that are expected to be experienced during the deployment or mission. It is expected that these projections will be generated during a pre-deployment/pre-mission maintenance-planning process.

■ Economics of Age Replacement

In addition to the calculation of deployment and mission risks, one can also compute a variety of economic-oriented metrics for each of the components covered by the life history-based prognostics system. Key metrics that one may want to compute are:

1. expected life remaining,
2. percentage of expected life remaining,
3. cost efficiency of an age-replacement policy (i.e., the ratio of expected component costs without an age-replacement policy to expected costs with such a policy),
4. expected combined costs per unit time of age replacements and in-service failures for a component maintained under an age-replacement policy,
5. replacement age that minimizes expected combined costs of age replacements and in-service failures,
6. average availability for a component maintained under an age-replacement policy,
7. replacement age that maximizes average availability for a component maintained under an age-replacement policy.

Mathematica functions for performing calculations such as the first two in the list above, can be found in TR-736. Current values for these two metrics should be continuously available to the operator, maintainer and maintenance planner. Functions for the other metrics pertaining to age replacement can be found in subsequent chapters of this report.

Optimal replacement ages, i.e., the fifth and seventh metrics above, can be obtained for any component that is subject to aging, provided the economic cost or system downtime due to in-service component failure is greater than the cost or downtime associated with planned component age replacement. The downtime and cost penalties associated with deployment or mission failure are generally greater than those due to planned age replacement. Sources of additional costs (compared to planned age replacement) due to in-service failures include:

- lost system usage (e.g., fuel, dollars per flight hour, wear & tear) due to an aborted mission,
- lost operator and crew time until system is repaired,
- high-priority shipping of parts,
- recovery personnel time, and
- fuel for, and wear & tear on, recovery vehicles.

One may choose to formulate an age-replacement policy based on component costs, in which case an optimal replacement age can be estimated that minimizes the combined component costs of in-service failure and planned age replacement.

Alternatively, one may choose to formulate an age-replacement policy based on system downtime. In this case, an optimal age-replacement policy can be estimated that maximizes availability considering the combined effects of system downtimes associated with in-service failure and planned age replacement. Sources of additional downtime (compared to planned age replacement) due to in-service failures include:

- time wasted on an aborted mission,

- time waiting for recovery,
- time ordering and waiting for parts,
- maintenance scheduling delay time, and
- maintenance travel time.

If the ratio of the cost of in-service failure to planned age replacement is the same as the ratio of corresponding downtimes, the same age-replacement policy will optimize both cost and availability.

As depicted in the Figure 2-1, performing these age-replacement computations requires data from the component databases. It is expected that age-replacement metrics, including optimal age-replacement policies, will be fed into the maintenance planning process in advance of a deployment or mission. Operators and maintainers could also be alerted to the approach of optimal component replacement ages via the on-board maintenance system.

▪ Pre-Deployment/Mission Maintenance Planning

It is expected that in advance of a deployment or mission, a maintenance-planning process should occur during which projections of the amount of usage the systems can be expected to receive may be generated. If a life history-based prognostics system is implemented, these projections can be used to calculate, and provide back to the maintenance planners, various metrics pertaining to deployment or mission risks, as discussed above. The prognostics system can also provide the maintenance planners with metrics concerning the economics of age replacement, as also discussed above. This will allow the maintenance planners to examine the metrics and map potential component replacement actions to the upcoming schedule. The planners can make decisions on component replacement actions based on either mission risks, economic considerations, or both.

It is important to consider both deployment/mission risks as well as economics. If one were to decide when to perform component age replacements based solely on mission risks, and the missions did not require much usage, age replacement may never be indicated even though a component may have only a small percentage of its life remaining and be at high risk of imminent failure. For this reason, the economics of planned component replacement should also be considered. The best approach may be to replace a component when first indicated by either considerations of deployment/mission risks or economics.

The costs of an in-service failure during a deployment or a mission may vary considerably based on the specific circumstances. For example, the cost of an in-service failure during a combat deployment may be much higher than during a training exercise. For this reason, it is expected that the maintenance planners may want to revise the costs and downtimes associated with in-service failures and age replacements in order to calculate risks and metrics tailored to the next deployment or mission.

Many age replacements will probably be accomplished before the deployment or mission. In cases where maintenance pauses are expected, the maintenance planners may want to pre-program alerts for the system operators or maintainers so they will automatically be notified of the need for component replacements during an upcoming maintenance pause. The maintenance planners would need to ensure that the parts and other resources required to perform the planned replacements will also be available.

- **Alerts**

The life history-based prognostics system should alert the driver, operator, or crew that a recommended component replacement action is approaching and the system should be brought in for maintenance. Perhaps a status display with components colored green, yellow, orange and then red would be most appropriate. The maintainer should also be alerted and advised to schedule the replacement and order the necessary parts. In addition, it may be helpful to allow the maintainers, and perhaps the operators, to input plans for the next portion of the life cycle and obtain a tailored set of component metrics and replacement recommendations from the prognostics system.

Conclusion

This chapter provided an initial conceptual overview concerning how one might implement life history-based prognostics on-board a system. This overview requires functions for performing two categories of computations. The first category is deployment/mission reliability computations and the second is age-replacement computations. Many of the functions required for deployment and mission reliability computations were previously developed and documented in TR-736. Functions that implement age-replacement economics are developed and documented in the remaining chapters of this report. The next chapter will present a collection of functions for implementing the economics of component age replacement.

Chapter 3

Economics of Component Age Replacement

Introduction

Our first step towards the implementation of a prognostics capability that is based on component life history was taken in AMSAA Technical Report 736 (TR-736). TR-736 addressed a variety of mission/deployment-risk metrics that could be computed at any point in the life cycle of a system, such as the probability that one of its critical components will fail during the next portion of the life-cycle, or the expected lifetime that the component has remaining. These metrics may be useful when one is deciding whether to replace a component in advance of a deployment or mission based on associated failure risks and/or the quantity of useful life that may be given up. One topic not addressed at that time was the relative cost of replacing the component before failure versus letting it operate until it fails. The purpose of this chapter is to address such component replacement costs in a detailed fashion.

Component Age Replacement

There is a class of maintenance policies found in the reliability literature termed *age replacement* policies. Age replacement means that a particular component on a system is replaced either when it reaches a specific age or fails in-service, whichever occurs first. Barlow and Proschan (1965) provide a theoretical treatment of various maintenance policies, including age-replacement policies, for a variety of situations. Gertsbakh (2000) is an up-to-date text on preventive-maintenance policies that will be more accessible to the practitioner.

Age-replacement policies are appropriate for components that meet two conditions:

1. The component ages (i.e., it has a failure rate or hazard function that increases with age or usage).
2. Failure of a component during actual operation (e.g., failure during a military deployment or mission) is more costly (i.e., in terms of financial cost or downtime) than replacing it at a pre-determined age under preferred circumstances.

The previous chapter listed several reasons why planned age replacement of a component is often advantageous in terms of both costs and downtime. If both of the above conditions are met, an age-replacement policy will reduce costs and increase availability. It is also possible to formulate optimal age-replacement policies (i.e., policies that minimize costs and/or maximize availability). The costs being referred to here are the aggregate costs of age replacement and in-service failures.

As will be seen later in the chapter, for components that are subject to aging, implementation of an age-replacement policy increases the reliability of the *socket*. For example, a policy of replacing a light bulb at a certain age does nothing to increase the reliability of the bulb but can dramatically increase the reliability of the socket the bulb occupies. (Age-replacement policies are not typically used with light fixtures, even though bulbs age, because planned age replacement of a bulb

typically offers little or no advantage compared to replacement at in-service failure.) Increasing the reliability of critical-component sockets increases the reliability of the system, subsystem or assembly that the socket is a part of. When the two conditions above are met, it is possible to determine replacement ages that minimize socket costs and/or maximize socket availability. This serves to reduce life cycle cost and increase the availability of the higher assemblies the component is a part of.

▪ Component Age-Replacement Package

A new package containing component age-replacement functions was developed in conjunction with this report and is provided as Appendix A. The new functions will be available for use when the package is loaded thus:

```
Needs["Reliability`ComponentAgeReplacement`"]
```

The current version of the package is:

```
? ComponentAgeReplacement
```

```
ComponentAgeReplacement.m (version 1.0) is a package that contains
  functions for analyzing and simulating component age-replacement policies.
```

This new package is intended to be used in conjunction with the package *Reliability`ConditionalDistributions`* that was developed in TR-736. Since it will also be needed it is also loaded:

```
Needs["Reliability`ConditionalDistributions`"]
```

The new functions defined in *Reliability`ComponentAgeReplacement`* are:

```
? Reliability`ComponentAgeReplacement`*
```

Reliability`ComponentAgeReplacement`

```
AgeReplacementEfficiency       AgeReplacementSimulation  MinLongTermCost
AgeReplacementMTBISF           ComponentAgeReplacement   ReliabilityBarChartData
AgeReplacementMTBISFBounds     LongTermAvailability      ReplacementAgeQ
AgeReplacementMTBR             LongTermCost              ReplacementCostQ
AgeReplacementReliability      MeanAgeReplacements
AgeReplacementReliabilityList  MeanDownTimeQ
```

The use of these functions in connection with deployment/mission-risk computations, as well as age-replacement economics, will be illustrated in the following subsections of this chapter as well as in the next chapter.

Age-Replacement Functions Useful for Deployment/Mission Risk Computations

Several new functions that are useful for deployment/mission-risk computations are:

- AgeReplacementReliability
- AgeReplacementMTBISF
- AgeReplacementMTBR
- MeanAgeReplacements.

Each will be discussed and illustrated in the following subsections.

■ Age-Replacement Reliability

A function which computes in-service reliability for specific replacement ages is `AgeReplacementReliability`:

```
? AgeReplacementReliability
```

```
AgeReplacementReliability[dist, T, t] is a function that calculates the reliability
  of a socket as a function of t with component age replacement occurring at T. The
  lifetime distribution of the component, assumed to have an increasing hazard function,
  is specified by dist. It is assumed that all components are new and have the same
  failure distribution. AgeReplacementReliability[dist, T, t, tprime] calculates the age-
  replacement reliability given the age of the initial component in the socket is tprime.
```

In Chapter 2 of TR-736, an actual track component with Weibull distribution parameters of 5.14 (shape) and 4602 (scale) was used for illustrative purposes. Since a Weibull shape parameter greater than 1 indicates aging is present, this track component was subject to rather strong aging. Assigning these parameter values to symbols to facilitate their subsequent use throughout this chapter:

```
wblshape = 5.14;
```

```
wblscale = 4602;
```

A plot of several age-replacement policies can be generated, as a function of in-service time, with `AgeReplacementReliability` thus (after first loading the standard add-on package *Graphics`Legend`* which provides functions for including legends):

```
Needs["Graphics`Legend`"]
```

```
Plot[{AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3500, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 4000, t],
   1 - CDF[WeibullDistribution[wblshape, wblscale], t]}, {t, 0, 10000},
  PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]}, {Hue[.2], Thickness[.005]},
    {Hue[.1], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
  Axes → False, Frame → True, FrameLabel → {"operating time, hours",
    "Socket Reliability", "Track Component Age-Replacement Reliability", None},
  PlotLegend → {"3,000", "3,500", "4,000", "None"}, LegendPosition → {1, -.4},
  LegendLabel → "Age Repl. (hrs)", LegendTextSpace → 2.1,
  LegendShadow → None, ImageSize → 72 * 8];
```

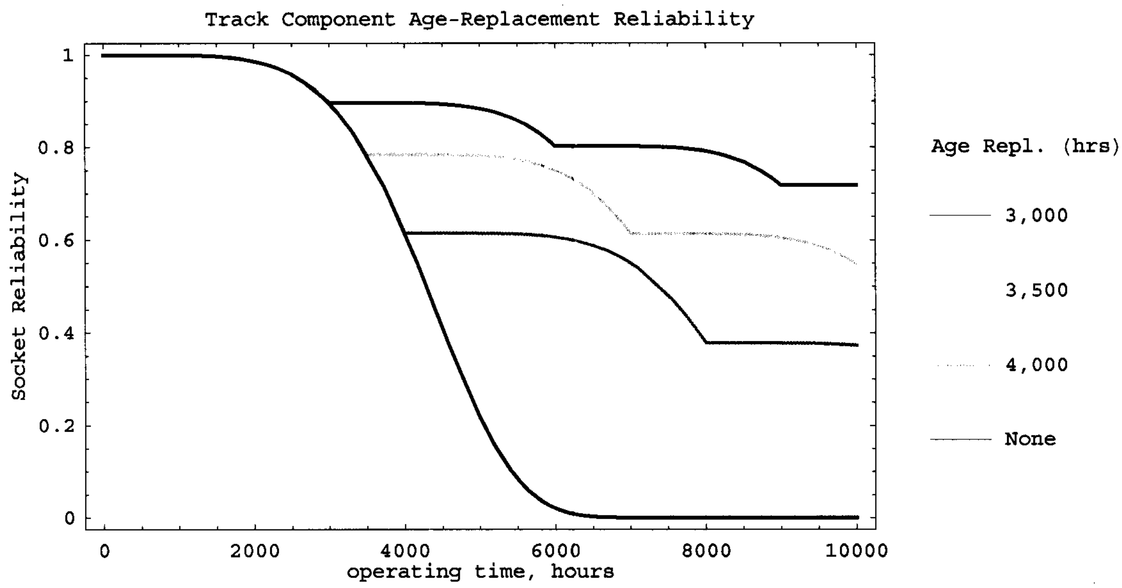

Figure 3-1

This figure depicts the probability that a socket will not have a component fail in-service before operating time *t*, assuming components are replaced at the specified age, or at in-service failure, whichever comes first. The green curve is the reliability given a component age-replacement policy of 3,000 hours. In essence, the green curve depicts the reliability of a socket that is to be occupied by a succession of components that are to be replaced every 3,000 hours, or at in-service failure, whichever occurs first. Curves are also displayed for replacement ages of 3,500 and 4,000 hours. The red curve represents the usual situation where components are only replaced after an in-service failure (i.e., a policy of no age replacement). There are sharp changes or discontinuities in the curves as each planned replacement age is encountered. Examination of the figure above reveals that shorter replacement ages serve to maintain reliability at a higher level by departing from the steep descent of the no age-replacement curve and jumping onto a new curve with little initial failure probability. In many cases, an optimal age-replacement policy can be computed (addressed later in the chapter).

It should also be noted that `AgeReplacementReliability` provides the reliability of a socket over a succession of age replacements, which is in contrast to the function in *Reliability`ConditionalDistributions`*, `ConditionalReliability`, which provides the reliability of a single, but not necessarily new component. If one is interested in looking at reliability for a large portion of the life cycle, or the entire life cycle, then `AgeReplacementReliability` is the appropriate choice since multiple component lifetimes are likely to be involved. One can examine the various age-replacement policies

depicted in figure 3-1 and determine the probability, given a particular age-replacement policy, that a single socket on a single system will have an in-service failure during the system life cycle or major portion thereof. `AgeReplacementReliability` will allow one to specify an age for the initial component in the socket, provided it is less than the replacement age. In the figure above, no such replacement age was specified which triggered an assumption that the initial component was new.

If one is interested in calculating the reliability of the current component for a small portion of the life cycle, and an age replacement is not being considered, or will not be encountered, then `ConditionalReliability` is probably the better choice. If the current age of the component is 2,000 hours, the reliability of the component from that point forward is:

```
Plot[ConditionalReliability[WeibullDistribution[wblshape, wblscale], t, 2000],
  {t, 2000, 7000}, PlotStyle → {Hue[.9], Thickness[.005]}, Axes → False, Frame → True,
  FrameLabel → {"operating time, hours", "Conditional Survival Probability", "Track
     Component Reliability Given Current Age 2000 Miles", None}, ImageSize → 72 * 6];
```

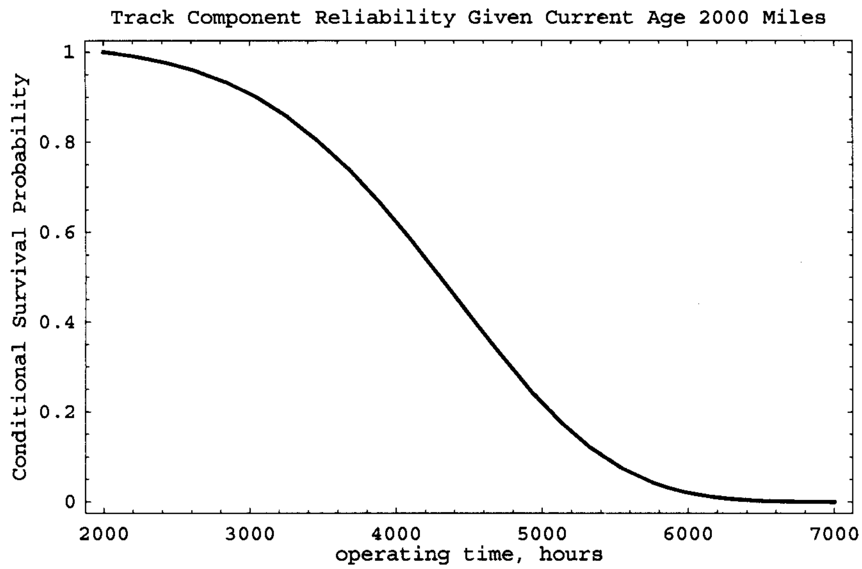

Figure 3-2

A survival probability from this figure could be used to help determine whether to replace the component before the deployment or mission. As discussed in Chapter 2, such a probability is one element that may trigger a pre-deployment, pre-deployment or between-mission replacement. In order to examine the impact of an age-replacement policy, one could generate a plot using `AgeReplacementReliability`. The impact of an replacement age of 3,000 hours is:

```
Plot[{AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3000, t, 2000],
   ConditionalReliability[WeibullDistribution[wblshape, wblscale], t, 2000]},
  {t, 2000, 7000}, PlotStyle → {{Hue[.2], Thickness[.005]}, {Hue[.9], Thickness[.005]}},
  Axes → False, Frame → True, FrameLabel → {"operating time, hours", "Socket Reliability",
    "Track Component Reliability Given Current Age 2000 Miles", None},
  PlotLegend → {"3,000", "None"}, LegendPosition → {1, -.4},
  LegendLabel → "Age Repl. (hrs)", LegendTextSpace → 2.1,
  LegendShadow → None, PlotRange → {0, 1}, ImageSize → 72 * 8];
```

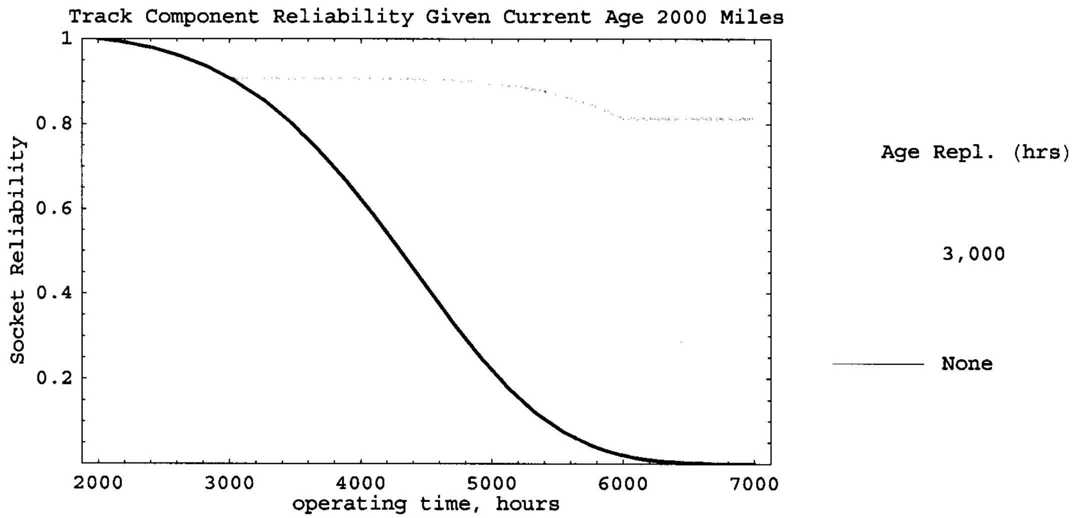

Figure 3-3

The reliability improvement due to age replacement is quite dramatic in the figure above.

As mentioned earlier in the chapter, age replacement is only beneficial when aging is present. If no aging is present, age replacement is not beneficial. `AgeReplacementReliability` will clearly depict this. If we reduce the Weibull shape parameter to one, the component will neither age nor improve. Figure 3-2 illustrates this case.

```
Plot[{AgeReplacementReliability[WeibullDistribution[1, wblscale], 3000, t],
   1 - CDF[WeibullDistribution[1, wblscale], t]}, {t, 0, 10000},
  PlotStyle → {{RGBColor[0, 1, 0], Dashing[{.15, .15}], Thickness[.005]},
    {RGBColor[1, 0, 0], Dashing[{0.1, 0.1}], Thickness[.005]}}, Axes → False,
  Frame → True, FrameLabel → {"operating time, hours", "Socket Reliability",
    "Age Replacement Reliability Given No Aging", None},
  PlotLegend → {"3,000 Hours", "None"}, LegendPosition → {1, -.4},
  LegendLabel → "Age-Replacement", LegendShadow → None, ImageSize → 72 * 8];
```

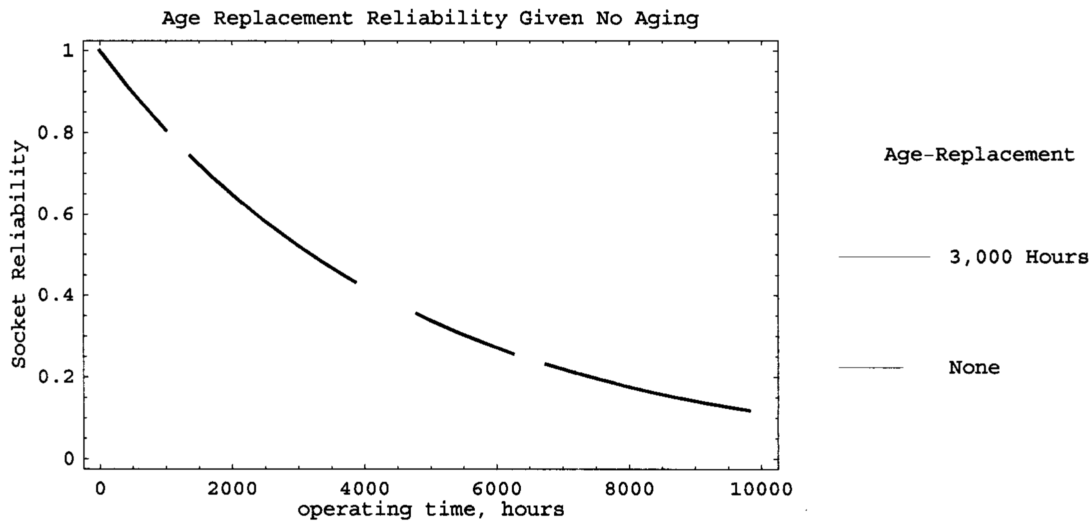

Figure 3-4

Two curves were plotted above, a green one for a 3,000 hour age replacement policy and a red one for no age replacement. The two curves plot as one indicating that when no aging is present, an age replacement policy does not improve reliability. If a component improves with age, use of an age-replacement policy will actually reduce in-service reliability. Reducing the Weibull shape parameter below one corresponds to a component that improves with age. This case is depicted in figure 3-4:

```
Plot[{AgeReplacementReliability[WeibullDistribution[.5, wblscale], 3000, t],
   1 - CDF[WeibullDistribution[.5, wblscale], t]}, {t, 0, 10000}, PlotStyle →
   {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
  Axes → False, Frame → True, FrameLabel → {"operating time, hours", "Socket Reliability",
    "Age-Replacement Reliability Given Improvement with Age", None},
  PlotLegend → {"3,000 Hours", "None"}, LegendPosition → {1, -.4},
  LegendLabel → "Age Replacement", LegendShadow → None, ImageSize → 72 * 8];
```

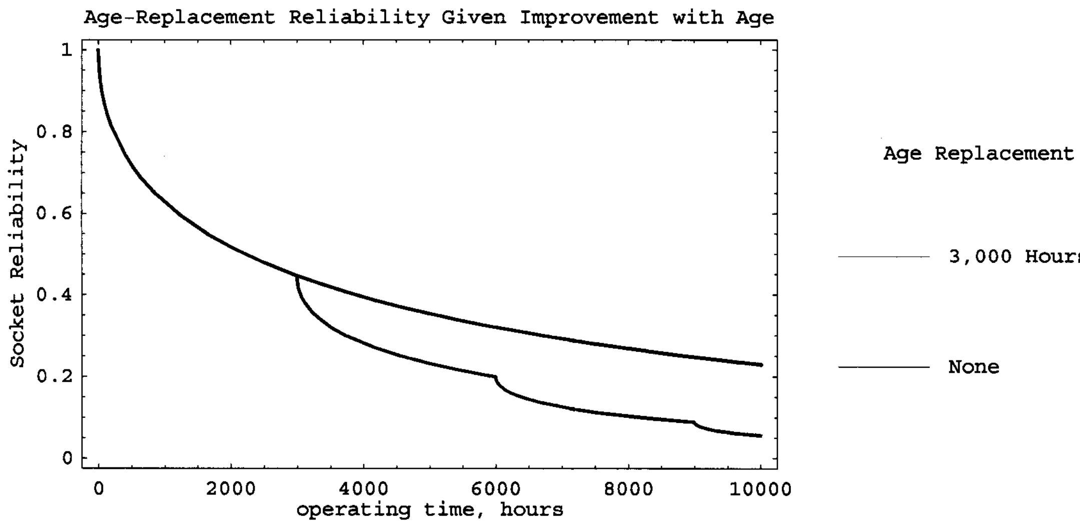

Figure 3-5

As can be readily seen, the age-replacement policy degrades reliability. Fortunately, components that improve with age are quite rare. When reliability that improves with age does occur, it is generally due to manufacturing defects or infant mortality which tend to occur during early life and are often driven out with process controls and/or burn in before the component ships.

■ Age-Replacement Mean Time Between In-Service Failures

The new function `AgeReplacementMTBISF` computes the expected or mean time between in-service failures (MTBISF) for a component that is replaced at a specified age:

```
? AgeReplacementMTBISF

AgeReplacementMTBISF[dist, T, opts] is a function that calculates the mean
  time between in-service failures of a component that is replaced at age T. The
  lifetime distribution of the component, assumed to have an increasing hazard
  function, is specified by dist. The optional argument opts specifies options to
  be used when integrating the lifetime cumulative distribution function such as
  assumptions regarding T and the distribution parameters. It is assumed that all
  components are new and have the same failure distribution. AgeReplacementMTBISF[
  WeibullDistribution[shape, scale], T] is a simplified form for the Weibull distribution.
```

This function requires the integration of the survivor function of the distribution. A specialized solution for the Weibull distribution, which obviates the need for integration, is also provided:

```
AgeReplacementMTBISF[WeibullDistribution[shape, scale], T]
```

$$\frac{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right]-\text{Gamma}\left[\frac{1}{\text{shape}},\left(\frac{T}{\text{scale}}\right)^{\text{shape}}\right]\right)}{\left(1-e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}\right)\text{shape}}$$

As can be seen, the solution above is closed-form except for the gamma function which is a built-in numerical function in *Mathematica*. We can now use this solution for computing and plotting the MTBISF for any component that is managed with an age-replacement policy and has a failure distribution that follows the Weibull distribution. We can plot the MTBISF for the track component as a function of replacement age thus:

```
Plot[{AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], T],
   Mean[WeibullDistribution[wblshape, wblscale]]},
  {T, 800, 6000}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", "Mean Time Between In-Service Failures",
    "Track Component Age-Replacement MTBISF", None}, PlotRange → All, PlotStyle →
   {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
  PlotLegend → {"MTBISF", "MTTF"}, LegendPosition → {1, -.4},
  LegendShadow → None, ImageSize → 72 * 8];
```

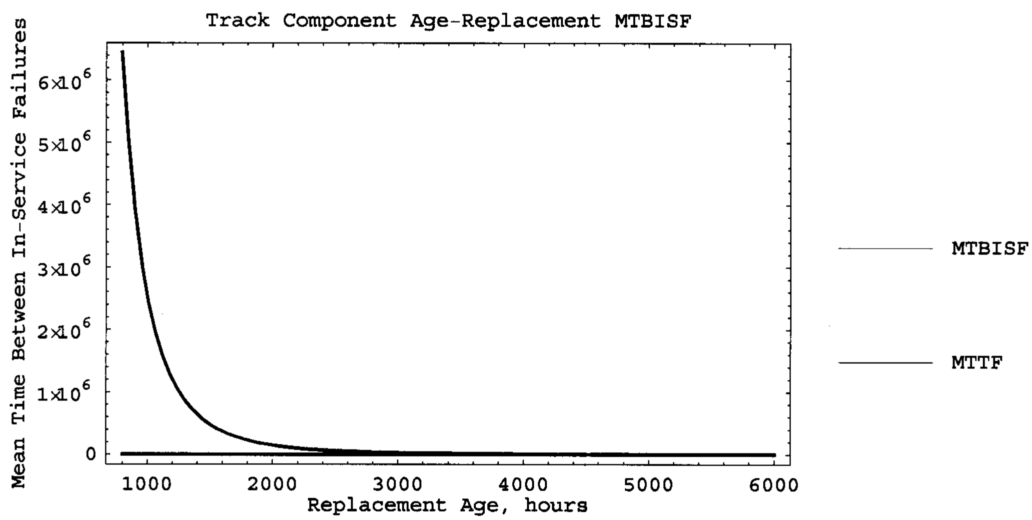

Figure 3-6

It should be noted that a lower limit of 800 hours was used for the replacement age in this plot, a value that is approximately 20% of the distribution mean. This is roughly comparable to the replacement age for a flight-critical component. Values lower than that are of little interest and will cause the y-axis to have extremely high MTBISF values that will obscure the region of interest. The MTBISF curve is green and the component mean time to failure (MTTF), which is the same as the MTBISF without age replacement, was plotted in red.

When the component is replaced early in its life, the MTBISF is very high. This is reasonable since replacements are occurring before much failure risk is incurred. As the replacement age increases, its benefit decreases, until eventually the MTBISF approaches the MTTF asymptotically from above. The scale for the MTBISF is still so large that it obscures more practical replacement ages, such as those in the vicinity of 3,000 hours. Restricting the replacement ages further we obtain:

```
Plot[{AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], T],
   Mean[WeibullDistribution[wblshape, wblscale]]},
  {T, 2000, 6000}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", "Mean Time Between In-Service Failures",
    "Track Component Age-Replacement MTBISF", None}, PlotRange → {0, 50000}, PlotStyle →
   {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
  PlotLegend → {"MTBISF", "MTTF"}, LegendPosition → {1, -.4},
  GridLines → Automatic, LegendShadow → None, ImageSize → 72 * 8];
```

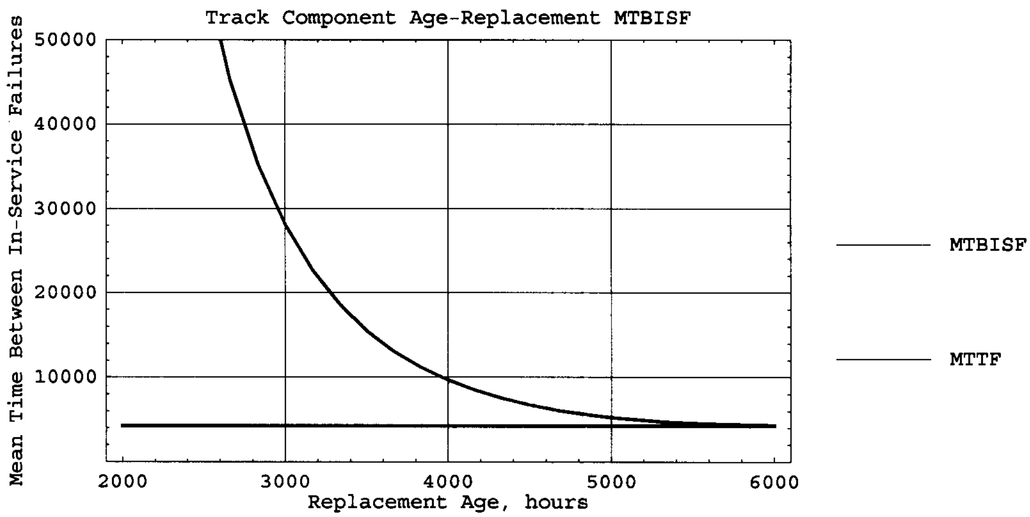

Figure 3-7

A policy of no age replacement can be viewed as replacement at an age of infinity. As one follows the MTBISF curve from right to left, it is clear that decreasing replacement ages below 6,000 hours begins to increase the MTBISF appreciably. Decreasing the replacement age below 3,000 hours has a very strong impact on MTBISF. A replacement age equal to the component MTTF (the MTTF is approximately equal to 4,200 hours) results in the MTBISF being a factor of 2 greater than the component MTTF. A replacement age of 3,000 causes the MTBISF to be a factor of 6 greater than the component MTTF.

There is little benefit to replacement ages above 6,000 since this component will almost certainly fail before that point. This can be verified by plotting the cumulative density function of the component:

```
Plot[CDF[WeibullDistribution[wblshape, wblscale], t], {t, 0, 6000},
  PlotStyle → {RGBColor[0, 0, 1], Thickness[.005]}, Axes → False, Frame → True,
  FrameLabel → {"operating time, hours", "Cumulative Failure Probability, CDF",
    "Track Component Failure Probability", None}, PlotRange → All, ImageSize → 72 * 6];
```

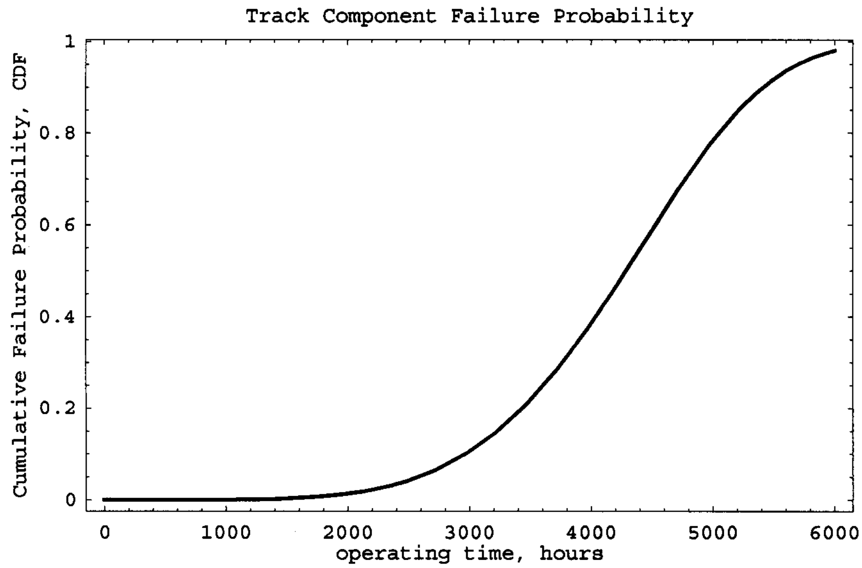

Figure 3-8

As expected, the component is highly likely to fail before 6,000 hours (i.e., the cumulative failure probability is approximately one).

The MTBISF curve provides valuable insight concerning how much the intervals between in-service failures will be increased by instituting an age-replacement policy for a component subject to aging. Without age replacement, one should expect an in-service failure of this component to occur, on average, every

```
Mean[WeibullDistribution[wblshape, wblscale]]

4232.15
```

hours. Instituting an age-replacement policy of 3,000 hours means that in-service failures of this component can be expected to occur every

```
AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], 3000]

28082.7
```

hours.

The insight provided by the MTBISF curve could prove quite valuable when determining how much an age-replacement policy will increase, on average, the intervals between in-service failures.

The MTBISF function could also be used in a different manner. If a system or subsystem has a predicted MTBF that is too low, the first choice is to improve the reliability of its least-reliable components by, for example, altering the design to reduce stresses. If that is not sufficient to meet end-item reliability requirements, one could manage those components

subject to aging with age-replacement policies. As we have seen, such policies can dramatically improve the component MTBISF and thus the system/subsystem reliability.

▪ Age-Replacement Mean Time Between Removals

The new function `AgeReplacementMTBR` computes the expected or Mean Time Between Removals (MTBR) for a component that is replaced at a specified age:

```
? AgeReplacementMTBR
```

```
AgeReplacementMTBR[dist, T, opts] is a function that calculates the mean time between
  removals (due either to failure or planned replacement) for a component that is
  replaced at age T. The lifetime distribution of the component, assumed to have an
  increasing hazard function, is specified by dist. The optional argument opts specifies
  options to be used when integrating the lifetime cumulative distribution function
  such as assumptions regarding T and the distribution parameters. AgeReplacementMTBR[
  AgeReplacementMTTF[WeibullDistribution[shape, scale], T] is a simplified form for
  the Weibull distribution. It is assumed that all components are new and have the
  same failure distribution. It is also assumed that replacement times are negligible.
  If not, AgeReplacementMTBR[dist, T, MDTf, MDTp, opts] or AgeReplacementMTBR[
  AgeReplacementMTTF[WeibullDistribution[shape, scale], T, MDTf, MDTp] can be used. MDTf
  is the mean downtime associated with the failure and replacement of a failed component.
  MDTp is the mean downtime associated with the planned replacement of a component.
```

It is important to recognize the distinction between `AgeReplacementMTBISF` and `AgeReplacementMTBR`. The former calculates the mean time between in-service failures whereas the latter calculates the mean time between removals. Removal is a broader term and includes both in-service failure and planned age replacement. Regardless of whether a component fails in-service or is removed because the replacement age is reached before failure, a removal occurs.

It is often assumed that the mean downtimes associated with in-service failure and with planned age replacement are negligible. Occasionally, it may not be appropriate to make that assumption. `AgeReplacementMTBR` will handle either case as its usage message above indicates.

This function requires the integration of the survivor function for the distribution. A specialized solution for the Weibull distribution, which obviates the need for integration, is also provided:

```
Simplify[AgeReplacementMTBR[WeibullDistribution[shape, scale], T]]
```

$$\frac{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right] - \text{Gamma}\left[\frac{1}{\text{shape}}, \left(\frac{T}{\text{scale}}\right)^{\text{shape}}\right]\right)}{\text{shape}}$$

We can now use this solution for computing and plotting the MTBR under an age-replacement policy. It may be helpful to plot this function for the track component on the same graph as the MTBISF:

```
Plot[{AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], T],
   Mean[WeibullDistribution[wblshape, wblscale]],
   AgeReplacementMTBR[WeibullDistribution[wblshape, wblscale], T]},
  {T, 2000, 6000}, Axes → False, Frame → True, FrameLabel → {"Replacement Age, hours",
    "Expected Time", "Track Component Age Replacement", None},
  PlotRange → {0, 50000}, PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]},
    {RGBColor[1, 0, 0], Thickness[.005]}, {RGBColor[0, 0, 1], Thickness[.005]}},
  PlotLegend → {"Btwn IS Fail", "To Fail", "Btwn Rem"}, LegendPosition → {.95, -.4},
  LegendLabel → "", LegendTextSpace → 2.1, LegendShadow → None, ImageSize → 72 * 8];
```

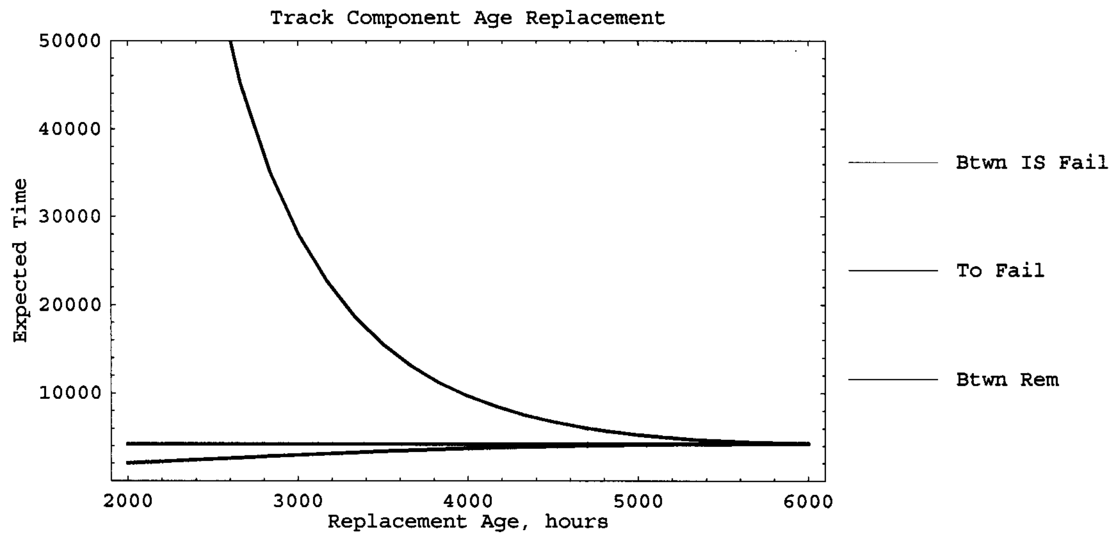

Figure 3-9

The MTBISF curve asymptotically approaches the MTTF line from above, whereas the MTBR curve approaches from below. It is appropriate that both the MTBISF and MTBF curves converge to the MTTF because as the replacement age goes to infinity, its benefits should disappear. It is appropriate that the MTBR curve should approach from below since it includes all of the in-service failures reflected in the MTTF line as well as the planned age replacements. One could say that the MTBR curve is the penalty that is paid in order to increase in-service reliability. The MTBR is not far below the MTTF whereas the MTBISF provides a large increase over the MTTF. This is another indication of the benefits of age replacement for this component.

■ Mean Age Replacements

The shorter the replacement age, the more age replacements one should expect to occur between in-service failures. `MeanAgeReplacements` is a function that will calculate this quantity:

```
?MeanAgeReplacements
```

```
MeanAgeReplacements[dist, T] is a function that calculates the number of
  age replacements expected to occur between in-service failures for a component
  replaced at age T. The lifetime distribution of the component, assumed to have
  an increasing hazard function, is specified by dist. It is assumed that all
  components are new and have the same failure distribution. MeanAgeReplacements[
  WeibullDistribution[shape, scale], T] is a simplified form for the Weibull distribution.
```

The expected number of age replacements can be plotted as a function of replacement age for the track component thus:

```
Plot[MeanAgeReplacements[WeibullDistribution[wblshape, wblscale], T],
  {T, 2000, 7000}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", "Mean Age Replacements Between Failures",
    "Track Component Mean Age Replacements", None}, PlotRange → Automatic,
  GridLines → Automatic, PlotStyle → {RGBColor[0, 1, 0], Thickness[.005]},
  ImageSize → 72 * 6];
```

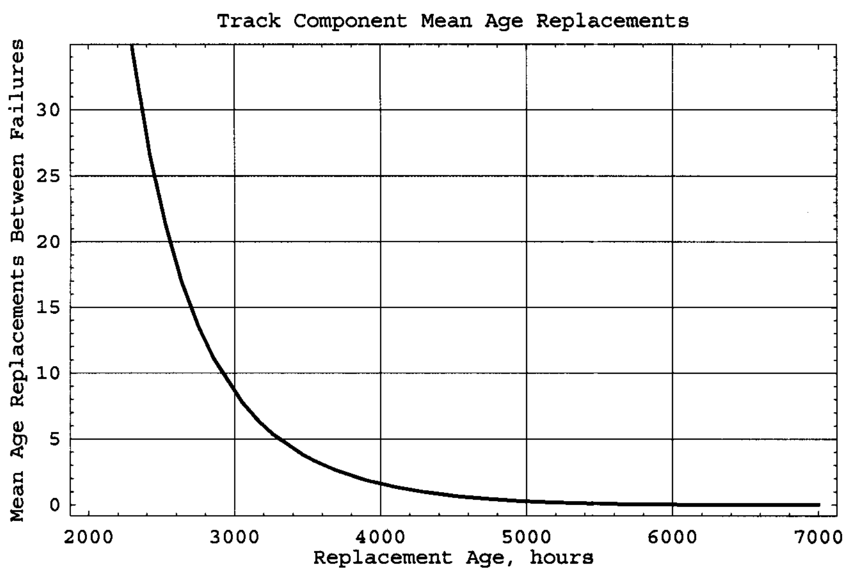

Figure 3-10

As one would expect, the mean number of age replacements approaches zero as the replacement age approaches infinity. For a replacement age of 3,000 hours, one should expect approximately

```
MeanAgeReplacements[WeibullDistribution[wblshape, wblscale], 3000]

8.52791
```

age replacements for each in-service failure.

It appears that the main use of the `MeanAgeReplacements` function is that it provides one with an idea as to the ratio of age replacements to in-service failures.

Functions Useful for Age-Replacement Economics

The functions presented thus far are primarily useful for calculating failure risks of aging components in advance of a deployment or mission. Consideration of these failure risks may trigger pre-deployment or pre-mission component replacement actions since corrective maintenance is generally much less convenient during deployments and missions. It may also be beneficial to take a longer view as well. When a component becomes less reliable with age, and the costs or downtime associated with in-service failure are greater than those due to planned age replacement, it is more economical to establish a replacement age for the component than to replace it only at failure. In fact, one can generally calculate optimal replacement ages for these components. It is true that removing a component before it fails results in the loss of a portion of its lifetime. But avoiding the cost penalty of an in-service failure can more than offset this. Optimization methods are available that balance the added costs of in-service failure against the lifetime lost due to a planned age replacement. These methods are frequently applied to industrial equipment since the financial consequences of failure are so great (e.g., the cost of shutting down a production line in the middle of a shift). In the remainder of this chapter, we will present functions useful for establishing age-replacement policies based on long-term economic considerations.

The cost function that one typically seeks to minimize with a component age-replacement policy is expected cost per unit time as time goes to infinity, which is expressed as:

$$\lim_{t\to\infty} \frac{C[t]}{t}$$

Costs can be in terms of either financial costs or downtime.

It is much easier to obtain an optimal replacement policy for an infinite time span than for a finite time span. The remainder of this chapter will consider optimization for an infinite time span. Functions useful for establishing age-replacement policies based on long-term economic considerations will be presented. The next chapter will consider optimization for a finite time span.

■ Minimizing Component Costs

The new function `LongTermCost` calculates the long-term expected cost per unit time:

```
? LongTermCost

LongTermCost[dist, T, costf, costp, opts] is a function that calculates the expected
  total cost per unit time as t approaches infinity for a component replaced at age
  T. The lifetime distribution of the component, assumed to have an increasing hazard
  function, is specified by dist. costf includes all costs incurred by the failure and
  replacement of a failed component. costp includes all costs associated with the planned
  replacement of a component. The optional argument opts specifies options to be used
  when integrating the lifetime cumulative distribution function such as assumptions
  regarding T and the distribution parameters. It is assumed that all components
  are new and have the same failure distribution. LongTermCost[WeibullDistribution[
  shape, scale], T, costf, costp] is a simplified form for the Weibull distribution.
```

The costs calculated by this function are the combined costs associated with the in-service failure and planned age replacement of a component. An in-service failure is generally more costly than a planned age replacement. Sources of additional

costs due to in-service failures include:

- lost system usage (e.g., fuel, dollars per flight hour, wear & tear) due to an aborted mission,
- lost operator and crew time until system is repaired,
- high-priority shipping of parts,
- recovery personnel time, and
- fuel for, and wear & tear on, recovery vehicles.

`LongTermCost` requires the integration of the survivor function of the distribution. A specialized solution for the Weibull distribution, which obviates the need for integration, is also provided:

`LongTermCost[WeibullDistribution[shape, scale], T, costF, costP]`

$$\frac{\left(\text{costP} + (\text{costF} - \text{costP})\left(1 - e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}\right)\right)\text{shape}}{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right] - \text{Gamma}\left[\frac{1}{\text{shape}}, \left(\frac{T}{\text{scale}}\right)^{\text{shape}}\right]\right)}$$

Let us consider the track component once again. In addition to the distribution parameters, we also need costs associated with in-service failure and planned age replacement of the component. Let us assume that the cost of an age replacement is $500 and the cost of an in-service failure is between $2,000 and $5,000. Before beginning to plot the long-term cost, it would be helpful to obtain a lower bound on the optimal long-term cost. The new function `MinLongTermCost` will allow us to calculate such a value:

`? MinLongTermCost`

```
MinLongTermCost[dist, costf, costp] is a function that calculates a replacement age that
  the solution to LongTermCost must be greater than. The lifetime distribution of the
  component, assumed to have an increasing hazard function, is specified by dist. costf
  includes all costs incurred by the failure and replacement of a failed component.
  costp includes all costs associated with the planned replacement of a component.
```

Should the cost of an in-service failure be $2,000, the minimum long-term cost would be:

`MinLongTermCost[WeibullDistribution[wblshape, wblscale], 2000, 500]`

```
1058.04
```

Should the cost of an in-service failure be $5,000, the minimum long-term cost would be:

`MinLongTermCost[WeibullDistribution[wblshape, wblscale], 5000, 500]`

```
423.215
```

We can now plot the expected cost per unit time, as a function of replacement age, using the smaller of these two minimums:

```
Plot[{LongTermCost[WeibullDistribution[wblshape, wblscale], T, 2000, 500],
                          2000
   ─────────────────────────────────────────────,
   Mean[WeibullDistribution[wblshape, wblscale]]
  LongTermCost[WeibullDistribution[wblshape, wblscale], T, 5000, 500],
                          5000
   ─────────────────────────────────────────────},
   Mean[WeibullDistribution[wblshape, wblscale]]
 {T, MinLongTermCost[WeibullDistribution[wblshape, wblscale], 5000, 500],
  2 * Mean[WeibullDistribution[wblshape, wblscale]]}, Axes → False, Frame → True,
 FrameLabel → {"Replacement Age, hours", "Expected Socket Cost ($) per Hour",
   "Track Component Long-Term Cost", None}, PlotRange → All,
 PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[1, 1, 0], Thickness[.005]},
   {RGBColor[0, 0, 1], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
 PlotLegend → {"2K", "2K, no AR", "5K", "5K, no AR"}, LegendPosition → {1, -.4},
 LegendLabel → "Failure Cost ($)", LegendShadow → None, ImageSize → 72 * 8];
```

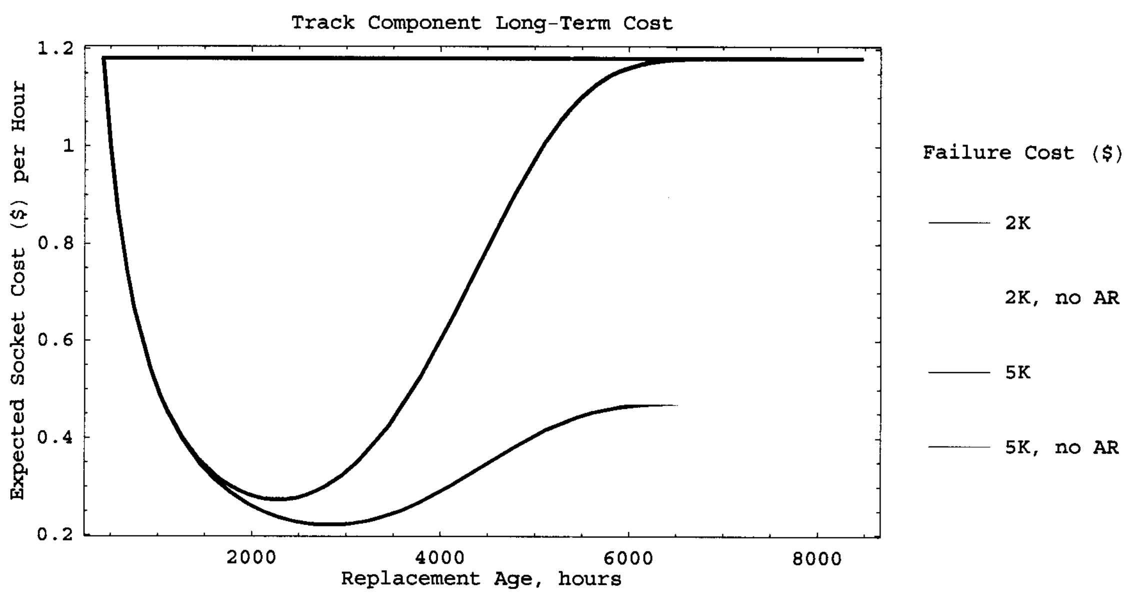

Figure 3-11

A value of twice the lifetime distribution mean was used as the upper limit since the replacement age is likely to be less than this provided there is a finite solution. The blue and green curves are the long-term socket costs given the $5,000 and $2,000 in-service failure costs, respectively. The yellow and red lines are the long-term costs that one would incur without age replacement, given in-service failure costs of $2,000 and $5,000, respectively. As the replacement age goes to infinity (i.e., as age replacement is turned off), the age-replacement long-term costs converge to the no-age replacement cost at approximately 6,000 hours. In both cases, the long-term costs can be dramatically reduced by an age-replacement policy. (This may not be the case when the aging is weak and/or the additional costs of in-service failure are small.) The long-term costs are minimized between replacement ages of 2,000 and 3,000 hours.

Comparison of the blue and green long-term cost curves reveals that an increased ratio of in-service failure cost to planned age-replacement cost has three effects:

1. Higher expected total costs,
2. expected costs are minimized at earlier replacement ages (i.e., the optimal replacement policy occurs at an earlier

replacement age), and

3. age replacement results in greater cost savings.

One always obtains curves of the type above when the component lifetime distribution has a hazard function that is continuous and strictly increasing, and the cost of failure is greater than the cost planned replacement. As can be readily seen, the expected long-term cost has a unique, finite minimum.

Let us narrow the replacement-age domain in order to better visualize the region where the expected costs are minimized:

```
Plot[{LongTermCost[WeibullDistribution[wblshape, wblscale], T, 2000, 500],
   LongTermCost[WeibullDistribution[wblshape, wblscale], T, 5000, 500]},
  {T, 2000, 3200}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", "Expected Socket Cost ($) per Hour",
    "Track Component Long-Term Cost", None}, PlotRange → All, PlotStyle →
   {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[0, 0, 1], Thickness[.005]}},
  PlotLegend → {"2,000", "5,000"}, LegendPosition → {1, -.4},
  LegendLabel → "Failure Cost ($)", LegendShadow → None, ImageSize → 72 * 8];
```

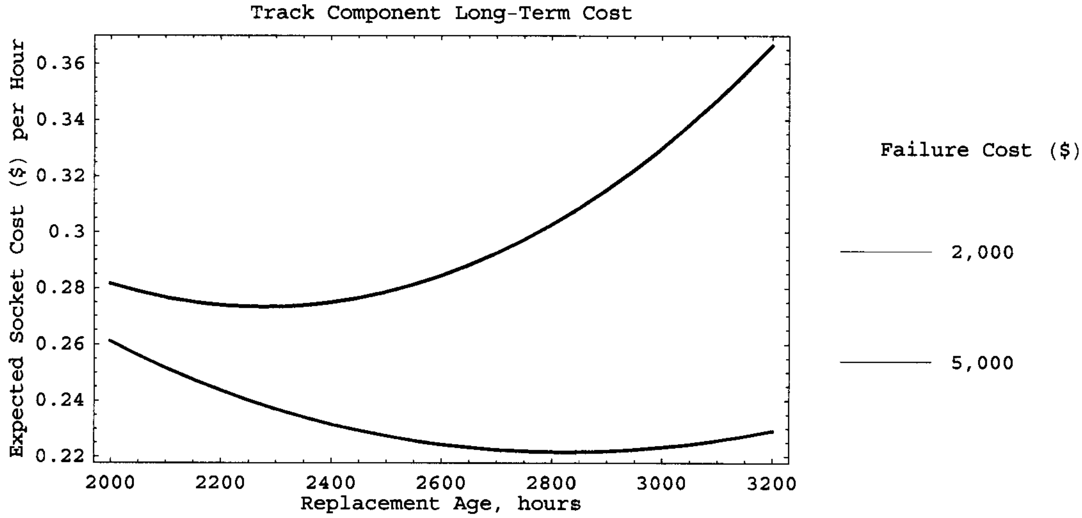

Figure 3-12

When the cost of in-service failure is $2,000, the expected cost is minimized in the vicinity of a replacement age of 2,800 hours. The built-in function `FindMinimum` can be used to find the optimal value:

```
FindMinimum[LongTermCost[WeibullDistribution[Rationalize[wblshape], wblscale],
  T, 2000, 500], {T, 2800}, WorkingPrecision → 22]

{0.2213771398998850778102, {T → 2822.499115235498568319}}
```

Given a failure cost of $2,000, the minimum expected cost of $0.22 per hour occurs at a replacement age of 2,822 hours. Extracting the optimal replacement age and assigning it as the value of *optage2000*:

```
optage2000 = T /. Last[%]

2822.499115235498568319
```

Without age replacement, the expected long-term socket cost would be:

```
                 2000
──────────────────────────────────────────
Mean[WeibullDistribution[wblshape, wblscale]]
```

```
0.472573
```

For this replacement age, one would expect

```
MeanAgeReplacements[WeibullDistribution[wblshape, wblscale], optage2000]
```

```
11.846
```

age replacements to occur between in-service failures.

The MTBISF for this age-replacement policy is:

```
AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], optage2000]
```

```
35789.5
```

The ratio of this MTBISF to the component MTTF without age replacement is:

```
AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], optage2000] /
 Mean[WeibullDistribution[wblshape, wblscale]]
```

```
8.45659
```

The MTBR, which includes both in-service failures and age replacements, is

```
AgeReplacementMTBR[WeibullDistribution[wblshape, wblscale], optage2000]
```

```
2786.05
```

which is just a small amount less than the replacement age. This is another indication that few in-service failures are to be expected if this replacement age is adopted.

One of the functions developed in TR-736, `ConditionalMeanLifeRemaining`, will allow us to calculate how much life a component is expected to still have at various ages. We can plot this function thus:

```
Plot[ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], age],
  {age, 0, 7000}, Axes → False, Frame → True, FrameLabel → {"age, hours",
    "Mean Life Remaining, hours", "Track Component Life Remaining", None},
  PlotStyle → {RGBColor[0, 0, 1], Thickness[.005]}, ImageSize → 72 * 6];
```

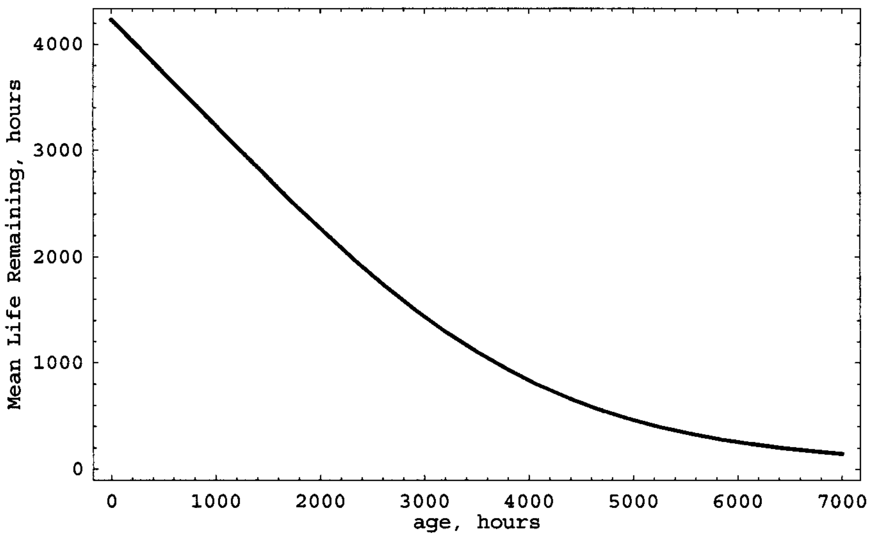

Figure 3-13

It is apparent from this plot that the component appears to have considerable expected lifetime left if it is replaced before 4,000 hours or so. The most cost-effective decision is to replace the component when it reaches an appreciably shorter age due to the greater costs associated with in-service failure. The lifetime remaining at the optimal replacement age when the cost of an in-service failure is $2,000 is:

```
ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], optage2000]

1568.17
```

The percentage of expected lifetime remaining at this replacement age compared to a new component is:

```
ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], optage2000] /
 ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], 0]

0.370538
```

If the failure cost is $5,000, it appears that the expected cost is minimized at approximately 2,300 hours. An accurate value can be found as before:

```
FindMinimum[LongTermCost[WeibullDistribution[Rationalize[wblshape], wblscale],
  T, 5000, 500], {T, 2300}, WorkingPrecision → 22]

{0.2731735984980963144524, {T → 2277.412563866978967444}}
```

Given a failure cost of $5,000, the minimum expected cost of $0.27 per hour occurs at a replacement age of 2,277 hours. Extracting the optimal replacement age and assigning it as the value of *optage5000*:

```
optage5000 = T /. Last[%]
```

2277.412563866978967444

Without age replacement, the expected long-term socket cost would be:

$$\frac{\mathbf{5000}}{\mathbf{Mean[WeibullDistribution[wblshape,\ wblscale]]}}$$

1.18143

For this replacement age, one would expect

```
MeanAgeReplacements[WeibullDistribution[wblshape, wblscale], optage5000]
```

36.681

age replacements to occur between in-service failures.

The MTBISF for this age-replacement policy is:

```
AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], optage5000]
```

85442.1

The ratio of this MTBISF to the component MTTF without age replacement is:

```
AgeReplacementMTBISF[WeibullDistribution[wblshape, wblscale], optage5000] /
 Mean[WeibullDistribution[wblshape, wblscale]]
```

20.1888

The MTBR is

```
AgeReplacementMTBR[WeibullDistribution[wblshape, wblscale], optage5000]
```

2267.51

which is just a small amount less than the replacement age which is appropriate since few in-service failures are to be expected if this replacement age is adopted.

The lifetime remaining at the optimal replacement age when the cost of an in-service failure is $5,000 is:

```
ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], optage5000]
```

2018.2

The percentage of expected lifetime remaining at this replacement age compared to a new component is:

```
ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], optage5000] /
 ConditionalMeanLifeRemaining[WeibullDistribution[wblshape, wblscale], 0]
```

0.476874

We can also generate an age-replacement reliability plot for these two policies as follows:

```
Plot[
  {AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], optage2000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], optage5000, t]},
  {t, 0, 10000}, PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]},
    {RGBColor[0, 0, 1], Thickness[.005]}}, Axes → False,
  Frame → True, FrameLabel → {"operating time, hours", "Reliability",
    "Track Component Age-Replacement Reliability", None},
  PlotLegend → {"2,000", "5,000"}, LegendPosition → {1, -.4},
  LegendLabel → "Failure Cost ($)", PlotRange → {0, 1},
  LegendShadow → None, ImageSize → 72 * 8];
```

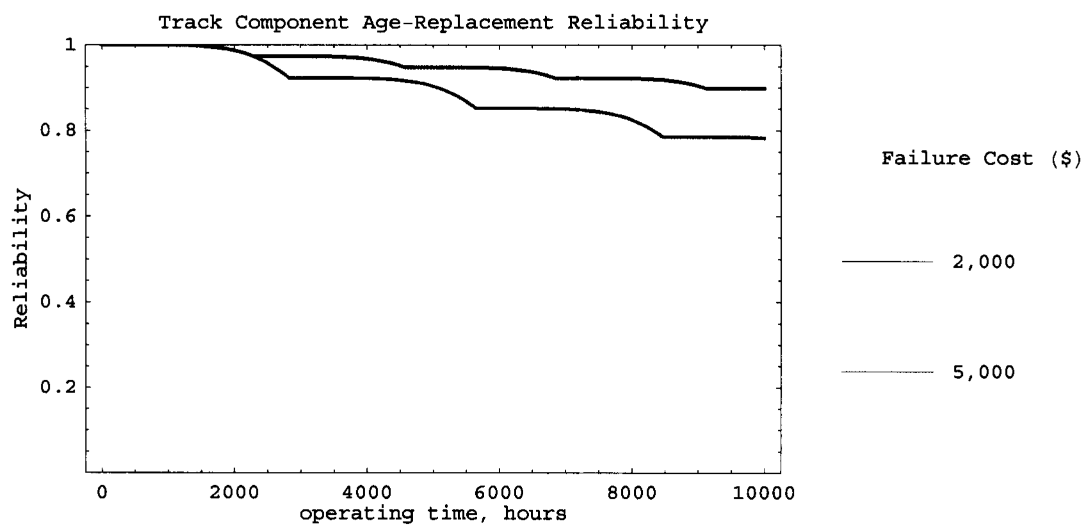

Figure 3-14

As this section illustrates, the function `LongTermCost` allows one to calculate an optimal replacement age and is particularly valuable.

■ Maximizing Maintenance Efficiency

A measure of the efficiency of an age-replacement policy is the ratio of the long-term cost without age-replacement to the cost with age-replacement. The new function `AgeReplacementEfficiency` calculates this measure:

```
? AgeReplacementEfficiency
```

AgeReplacementEfficiency[dist, T, costf, costp, opts] is a function that calculates the expected total cost per unit time for a component without age replacement divided by the expected total cost per unit time with age replacement. This is a value expected over the long term. The lifetime distribution of the component, assumed to have an increasing hazard function, is specified by dist. costf includes all costs incurred by the failure and replacement of a failed component. costp includes all costs associated with the planned replacement of a component. The optional argument opts specifies options to be used when integrating the lifetime cumulative distribution function such as assumptions regarding T and the distribution parameters. It is assumed that all components are new and have the same failure distribution. AgeReplacementEfficiency[WeibullDistribution[shape, scale], T, costf, costp] is a simplified form for the Weibull distribution.

If the hazard function is continuous and strictly increasing, and the cost of failure is greater than the cost of planned replacement, the age-replacement efficiency will be greater than one. Taking the track component once again, as well as the in-service failure and planned age replacement costs assumed in the previous subsection, we can plot the efficiency as a function of replacement age thus:

```
Plot[{AgeReplacementEfficiency[WeibullDistribution[wblshape, wblscale], T, 2000, 500],
   AgeReplacementEfficiency[WeibullDistribution[wblshape, wblscale], T, 5000, 500]},
  {T, MinLongTermCost[WeibullDistribution[wblshape, wblscale], 5000, 500],
   2 * Mean[WeibullDistribution[wblshape, wblscale]]}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", " Age Replacement Efficiency",
    "Track Component Age Replacement", None}, PlotRange → All, PlotStyle →
   {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[0, 0, 1], Thickness[.005]}},
  PlotLegend → {"2,000", "5,000"}, LegendPosition → {1, -.4},
  LegendLabel → "Failure Cost ($)", LegendShadow → None, ImageSize → 72 * 8];
```

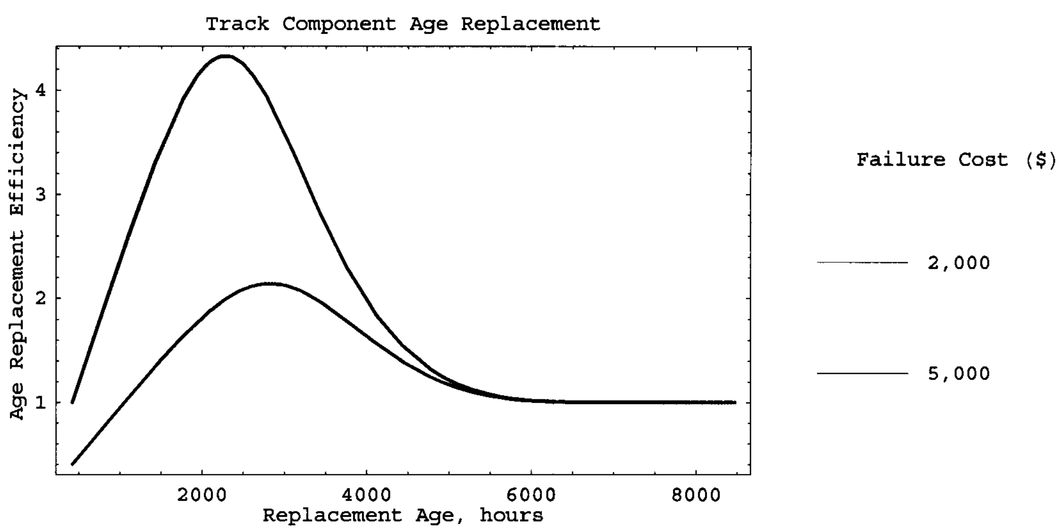

Figure 3-15

An efficiency greater than one occurs when the component ages and the cost of failure is greater than the cost of planned replacement. In this case, one saves money by performing age replacement. The efficiency is maximized at the optimal replacement age. Using the optimal replacement age for the case where the failure cost is $2,000, the efficiency is:

```
AgeReplacementEfficiency[
 WeibullDistribution[Rationalize[wblshape], wblscale], optage2000, 2000, 500]
```

```
2.134696355804374845 39
```

Using the optimal replacement age for the case where the failure cost is $5,000, the efficiency is:

```
AgeReplacementEfficiency[
 WeibullDistribution[Rationalize[wblshape], wblscale], optage5000, 5000, 500]
```

```
4.3248412035503972603
```

The greater the ratio of failure cost to planned-replacement cost, the greater the efficiency that age replacement provides. The other effect that increases efficiency is when the lifetime distribution is more peaked about the MTTF. For the Weibull distribution, this means that the greater the shape parameter, the greater the age-replacement efficiency.

By maximizing the age-replacement efficiency, we will obtain the same optimal replacement age as we did by minimizing the long term cost.

```
FindMaximum[
 AgeReplacementEfficiency[WeibullDistribution[Rationalize[wblshape], wblscale],
  T, 2000, 500], {T, 2800}, WorkingPrecision → 22]
```

```
{2.134696355804374845393, {T → 2822.499115235491064836}}
```

This is the same optimal replacement age as obtained previously with the function `LongTermCost`. `AgeReplacement-Efficiency` is a normalized measure that one may also choose to consider when formulating an age-replacement policy.

■ Maximizing Availability

The last new function to be presented in this chapter is `LongTermAvailability`:

```
? LongTermAvailability
```

```
LongTermAvailability[dist, T, MDTf, MDTp, opts] is a function that calculates the
  expected availability as t approaches infinity for a component replaced at age T.
  The lifetime distribution of the component, assumed to have an increasing hazard
  function, is specified by dist. MDTf is the mean downtime associated with the failure
  and replacement of a failed component. MDTp is the mean downtime associated with the
  planned replacement of a component. The optional argument opts specifies options to be
  used when integrating the lifetime cumulative distribution function such as assumptions
  regarding T and the distribution parameters. It is assumed that all components are
  new and have the same failure distribution. LongTermAvailability[WeibullDistribution[
  shape, scale], T, MDTf, MDTp] is a simplified form for the Weibull distribution.
```

The functions `LongTermCost` and `AgeReplacementEfficiency` allowed us to find the replacement age that minimizes the aggregate of in-service and age-replacement costs. Age replacement will reduce long-term costs provided the component becomes less reliable with age and in-service failures are more costly than age-replacement failures. One may also choose to consider steady-state availability. Instead of financial cost, we now consider downtime. The downtime resulting from in-service failures is generally greater than that due to planned age replacements. Sources of additional downtime due to in-service failures include:

- time wasted on an aborted mission,
- time waiting for recovery,
- time ordering and waiting for parts,
- maintenance scheduling delay time, and
- maintenance travel time.

Before plotting long-term availability as a function of replacement age, let us determine what the availability that would occur without age replacement is. We can determine this by setting the replacement age to infinity which effectively turns age replacement off:

```
Simplify[LongTermAvailability[dist, Infinity, mdtf, mdtp]]
```

$$\frac{\text{Mean[dist]}}{\text{mdtf} + \text{Mean[dist]}}$$

We will include the long-term availability without age replacement on the availability figure.

When considering financial costs, we assumed that the cost of an in-service failure was between four and ten times that of a planned age replacement. To be consistent, let us also assume that the mean down time associated with age replacement is 1 hour, and for in-service failures it is between 4 and 10 hours. We can generate an availability plot as follows:

```
Plot[{LongTermAvailability[WeibullDistribution[wblshape, wblscale], T, 4, 1],
    Mean[WeibullDistribution[wblshape, wblscale]]
   ---------------------------------------------- ,
   4 + Mean[WeibullDistribution[wblshape, wblscale]]
   LongTermAvailability[WeibullDistribution[wblshape, wblscale], T, 10, 1],
    Mean[WeibullDistribution[wblshape, wblscale]]
   ----------------------------------------------- }, {T, 500, 6000},
   10 + Mean[WeibullDistribution[wblshape, wblscale]]
  Axes → False, Frame → True, FrameLabel → {"Replacement Age, hours",
    "Socket Availability", "Track Component Long-Term Availability", None},
  PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]}, {RGBColor[1, 1, 0], Thickness[.005]},
    {RGBColor[0, 0, 1], Thickness[.005]}, {RGBColor[1, 0, 0], Thickness[.005]}},
  PlotLegend → {"4", "4, no AR", "10", "10, no AR"}, LegendPosition → {1, -.4},
  LegendLabel → "MDT Ratio", LegendShadow → None,
  PlotRange → {.995, 1}, ImageSize → 72 * 8];
```

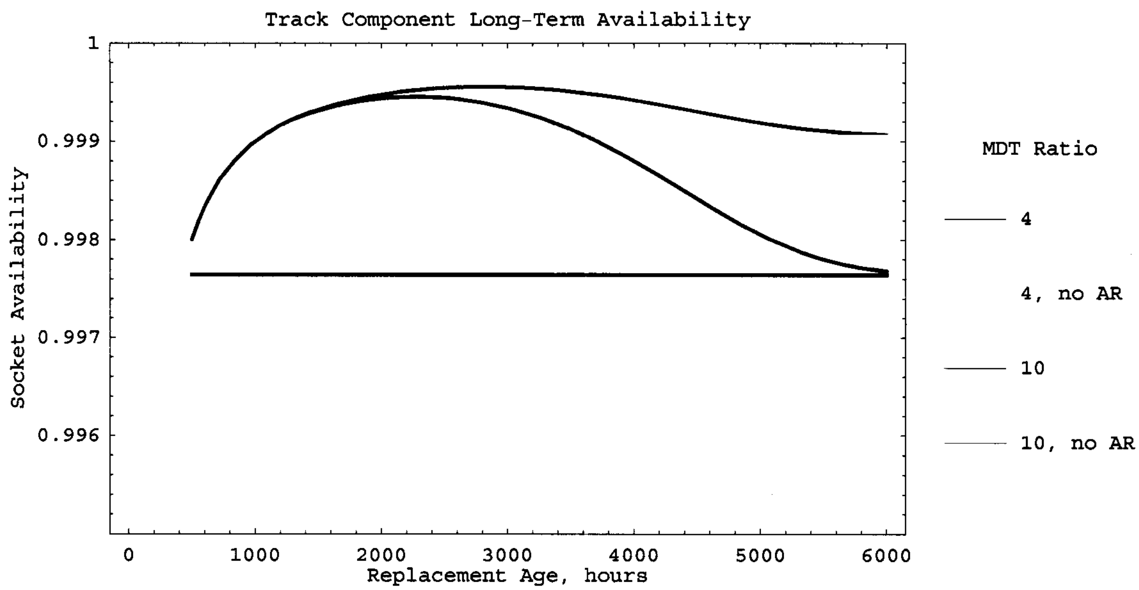

Figure 3-16

As we can see, the long-term availability of this component is improved by age replacement. The optimal ages occur between 2,000 and 3,000 hours. As the replacement age increases beyond 6,000 hours, each age-replacement availability curve approaches the availability line associated with a no age-replacement policy, as it should. Since we assumed mean downtime values for in-service failure and age replacement that have the same ratio as the corresponding cost values, maximizing the long-term availability will provide the same replacement age.

```
FindMaximum[
 LongTermAvailability[WeibullDistribution[Rationalize[wblshape], wblscale], T, 4, 1],
 {T, 2800}, WorkingPrecision → 22]

{0.9995574416647972019445, {T → 2822.499115235498567855}}
```

This is the same optimal replacement age that we obtained with `LongTermCost` for this case.

The function `LongTermAvailability` permits one to examine the impact of an age-replacement policy on availability. It should be remembered that this is just the portion of the system/subsystem availability due to this component (i.e., the availability of the socket the component occupies). In some cases, long-term cost may be the most important factor, particularly if the availability improvement provided by age replacement is not very great. We assumed here a ratio of downtimes due to in-service failures and age replacements that was identical to the ratio of costs. In general, this will not be the case and different optimal replacement ages will be obtained when optimizing financial costs and availability. It would be best to obtain replacement ages both ways, see how different they are, and rely on whichever is most important.

Summary

In this chapter, a new *Mathematica* add-on package that defines a collection of functions useful for formulation of age-replacement policies was presented. The use of each of the key functions was illustrated. Processes for obtaining age-replacement policies that are optimum with respect to either financial cost or availability were also presented. These component age-replacement functions complement the functions developed in TR-736, which were mainly focused on computing risks of component failure before a deployment or mission. The age-replacement functions permit one to consider the long-term economic or availability consequences of establishing a component age-replacement policy. Use of the functions developed in TR-736, in conjunction with the functions developed herein, will allow one to consider both the mission/deployment risks as well as the long-term benefits of replacing an aging component before it fails.

The functions in this chapter that allow one to determine optimal replacement ages assumed an infinite time horizon (i.e., assumed $t \to \infty$). Strictly speaking, this may be too long of a time horizon but it simplifies the mathematics considerably. In the next chapter, a simulation will be used to obtain optimal replacement ages for a finite time horizon.

Another assumption made when optimizing replacement ages in this chapter was that the currently-installed component was new. The simulation in the next chapter, in addition to avoiding the infinite-time-horizon assumption, will also allow us to avoid this assumption. (The functions `AgeReplacementMTBISF`, `AgeReplacementMTBR` and `MeanAgeReplacements` also assume that all components are new, but since these metrics capture the long-term impact of an age-replacement policy, this assumption is reasonable.)

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Simulation of Component Age Replacement

Introduction

Two assumptions were made in the last chapter in connection with the calculation of optimal replacement age. First, an infinite time horizon was assumed. The combined cost of in-service failure and age replacement was optimized over an infinite time horizon (i.e., as operating usage approached infinity). Availability was also optimized over an infinite time horizon. The assumption of an infinite time horizon simplifies the mathematics considerably. If one is only concerned with large portions of the life cycle, this assumption is likely to be reasonable. If one, however, is interested in small segments of the life cycle, this may not be the case. The second assumption was that the initial component in the socket is new. Once again, if one is only concerned about large segments of the life cycle, this is probably a reasonable assumption. For smaller segments of the life cycle (e.g., an upcoming deployment) it would be preferable to calculate the optimal replacement age conditioned on the current age of the installed component. In this chapter, we will obtain and evaluate approximately-optimal replacement ages through a simulation over a finite interval with a currently-installed component that is not necessarily new.

Simulation of Reliability

In Chapter 3, the age-replacement reliability of a socket was calculated and plotted as a function of time. The currently-installed component, as well as all of the replacement components, were assumed to be new in order to be consistent with the age-replacement optimization sections. (The new function `AgeReplacementReliability` will accept a currently-installed component that is not new.) In addition, one can use the age-replacement simulation functions defined in the add-on package *Reliability`ComponentAgeReplacement`* to consider the impact if the current component is not new.

```
Needs["Reliability`ComponentAgeReplacement`"]
```

The new function `AgeReplacementSimulation` will generate either reliability or cost simulations of an age-replacement policy:

```
? AgeReplacementSimulation
```

```
AgeReplacementSimulation[dist, T, intervalEnd, tprime, trialQty] is a function that performs
  a socket-reliability simulation of an age-replacement policy using a replacement age
  of T for the specified distribution. The age of the original component is tprime and
  the component will be replaced at T, or upon in-service failure, until intervalEnd
  occurs. The number of simulation trials is specified by trialQty. It is assumed
  that all replacement components are new and have the same failure distribution as
  the currently-installed component. AgeReplacementSimulation[dist, T, costf, costp,
  intervalEnd, tprime, trialQty] performs an cost simulation for an age-replacement
  policy and provides an expected socket cost for the simulated interval given the
  cost of an age-replacement is costp and the cost of an in-service failure is costf.
```

Consequently, two functions, `AgeReplacementReliability` and `AgeReplacementSimulation` are available for calculating age-replacement reliability. In addition to enabling double-checking, the age-replacement reliability simulation generates useful statistics conditioned on the age of the current component. Also, the age-replacement reliability simulation was a necessary part of simulation-based, age-replacement optimization for finite time intervals.

We will continue to use the track component analyzed in both AMSAA Technical Report 736 and Chapter 3 of this report. Assigning the Weibull parameters for it to symbols will facilitate subsequent illustrations throughout this chapter:

```
wblshape = 5.14;

wblscale = 4602;
```

In Chapter 3, it was determined that when the cost of age replacement and in-service failure are $500 and $2,000, respectively, the optimal replacement age is 2,822 hours. Let's simulate this replacement age for 10,000 hours of operating time. We will first seed the pseudorandom number generator in order to get repeatable simulation results.

```
SeedRandom[1]
```

Now we will simulate 100,000 sockets for 10,000 operating hours each assuming that the age of the currently-installed component is 2,000 hours:

```
Shallow[simlist1 = AgeReplacementSimulation[
   WeibullDistribution[wblshape, wblscale], 2822, 10000, 2000, 100000]]

{{3, 0, Null, 4239.96}, {3, 0, Null, 4073.08}, {2, 1, 5449.66, 4413.45},
 {3, 0, Null, 4519.}, {3, 0, Null, 6680.53}, {3, 0, Null, 1711.75}, {3, 0, Null, 5605.6},
 {3, 0, Null, 5546.23}, {3, 0, Null, 5922.72}, {2, 1, 8305.15, 4112.01}, <<99990>>}
```

The simulation provides a list of quadruples, one for each iteration. The elements of each quadruple, in order, are the number of age replacements, the number of in-service failures, the time to the first in-service failure and the lifetime lost due to age replacement(s). Before plotting socket reliability as a function of operating time, we must extract and manipulate the data on the times-to-first-in-service failure. The new function `ReliabilityBarChartData` will manipulate the simulation data so that we can generate a reliability histogram using the `GeneralizedBarChart` function defined in the standard add-on package *Graphics`Graphics`*:

```
? ReliabilityBarChartData
```

```
ReliabilityBarChartData[simdata, simEnd, barQty] is a function that takes
  a list containing age-replacement simulation results (simdata) and generates
  a list of triples suitable for GeneralizedBarChart. The end of the simulation
  interval used in the age-replacement simulation is specified by simEnd. The
  quantity of intervals to be generated for the bar chart is specified by barQty.
```

Manipulating the data with `ReliabilityBarChartData` first:

```
N[bardata = ReliabilityBarChartData[simlist1, 10000, 50]];
```

Before using `GeneralizedBarChart`, the standard add-on package *Graphics`Graphics`* must be loaded:

```
Needs["Graphics`Graphics`"]
```

Now the histogram is plotted:

```
barplot = GeneralizedBarChart[bardata, Axes → False, Frame → True,
   FrameLabel → {"operating time, hours", "Socket Reliability",
     "Track Component Age-Replacement Simulation", None}, ImageSize → 72 * 6];
```

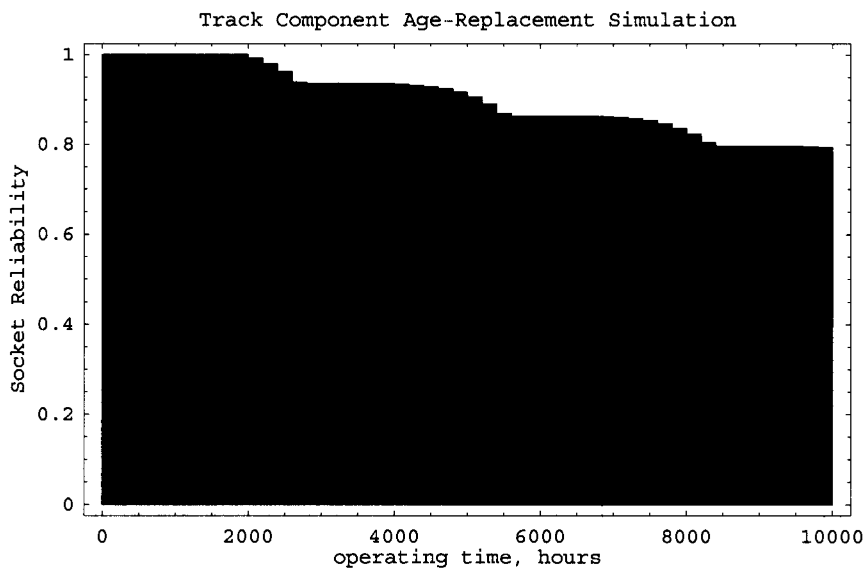

Figure 4-1

A comparable plot can be generated with the function `AgeReplacementReliability` (introduced in Chapter 3) thus:

```
funplot = Plot[AgeReplacementReliability[
    WeibullDistribution[wblshape, wblscale], 2822, t, 2000], {t, 0, 10000},
   PlotStyle → {Thickness[.005], RGBColor[0, 1, 0]}, PlotRange → {.55, 1}, Axes → False,
   Frame → True, FrameLabel → {"operating time, hours", "Socket Reliability",
     "Track Component Age-Replacement Reliability", None}, ImageSize → 72 * 6];
```

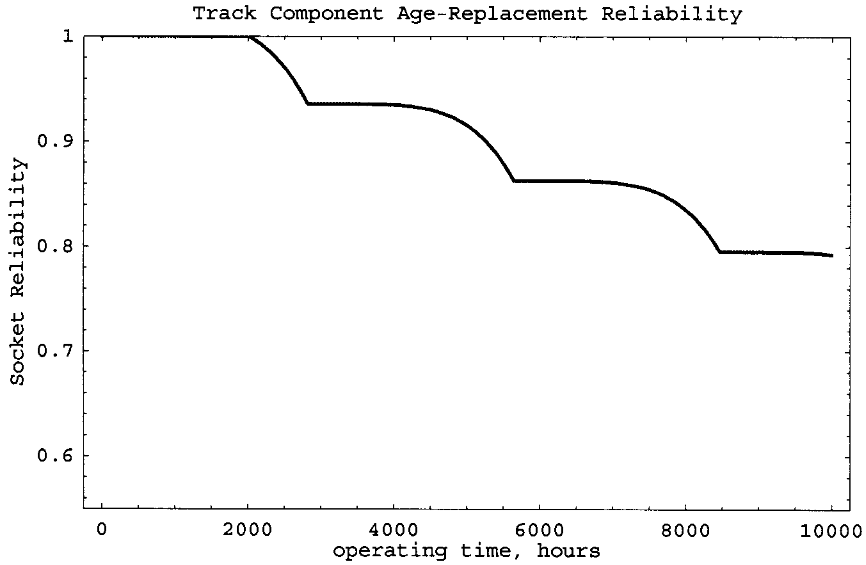

Figure 4-2

The two plots can now be overlaid:

```
Show[barplot, funplot];
```

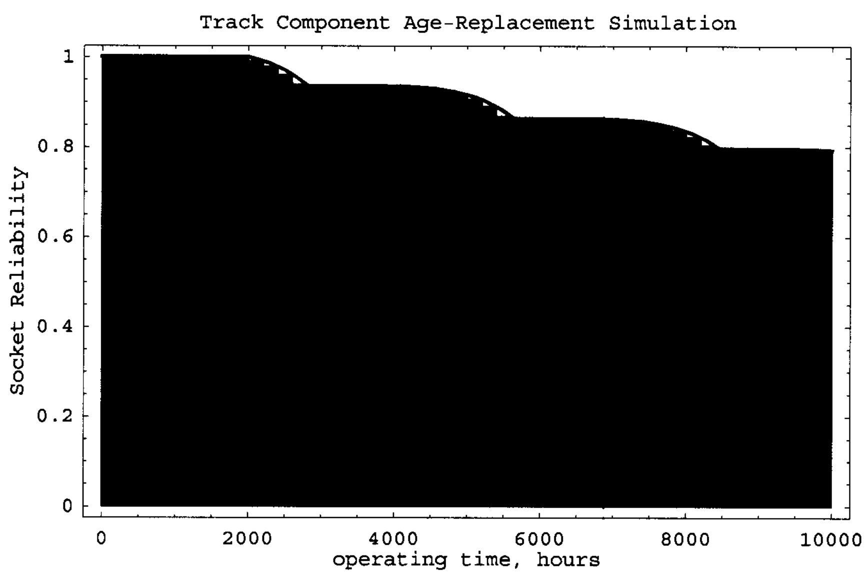

Figure 4-3

The two plots show excellent agreement.

Short-Term Optimal Replacement Age

Regardless of whether a system is facing a deployment or other important life-cycle phase, if the component ages and the cost of an in-service failure is ordinarily appreciably greater than the cost of an age replacement, establishing a long-term, age-replacement policy will reduce socket costs. Even when a long-term, age-replacement policy exists, short-term policies may be advisable if the cost difference is going to be increased further still.

In this section, a series of simulations will be run in order to approximately determine and evaluate optimal replacement ages for finite time intervals. The track component will be used once again. The objective is to illustrate the process using `AgeReplacementSimulation`.

In the finite time span models, one minimizes expected socket cost during a specific interval. The expected cost equation for the interval $[t_1, t_2]$ used in `AgeReplacementSimulation` is that recommended by Barlow and Proschan [1996, p. 85, equation 2.1]:

$$C(t) = c_f\, EN_f(t) + c_p\, EN_p(t)$$

where:

- c_f is the expected cost of component failure and includes all costs associated with failure and replacement of the component,
- c_p is the expected cost of a planned component replacement, and $c_p < c_f$,
- $EN_f(t)$ is the expected number of failures during $[t_1, t_2]$,
- $EN_p(t)$ is the expected number of planned replacements of non-failed items during $[t_1, t_2]$.

The primary situation in which one may want to consider developing a short-term, optimal replacement age for a component that ages with use is when the cost of an in-service failure is not typically much greater than an age replacement. But a deployment, mission, exercise or some other life-cycle segment is approaching, and the cost of an in-service failure will be much greater than the cost of an age replacement. (It is assumed that there will be opportunities within the life-cycle segment during which an age replacement could be performed, if desired.) We will now consider a few such cases.

■ Case 1: 3,000 Hour Interval, Currently-Installed Component is New

Let us use the track component from Chapter 3 and consider the following hypothetical case. The typical cost during the life cycle of the component for either an age replacement and an in-service failure is $500. A deployment is approaching and the cost of an in-service failure will increase to $2,000. It is anticipated that the system will be operated for 3,000 hours during the deployment. If the currently-installed component is new, does a short-term, age-replacement policy make sense? Let's simulate a range of replacement ages for 3,000 hours of additional operating time and compare the expected socket costs. We will assign the desired quantity of simulation trials as the value of the symbol *trials* since we will be using this value in all the cases:

```
trials = 250000;
```

The assumed age of the currently-installed component for this case is assigned as the value of the symbol *age* thus:

```
age = 0;
```

Now we will simulate replacement ages of 999, 1001, 1499, 1501, 2000, 2500, and 2999 hours:

```
TableForm[Round[simpts =
   Map[({#, AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale], #, 2000,
        500, age + 3000, age, trials]}) &, {999, 1001, 1499, 1501, 2000, 2500, 2999}]],
 TableHeadings → {None, {"Replacement Age", "Socket $ for 3,000 Hours"}},
 TableAlignments → Center]
```

Replacement Age	Socket $ for 3,000 Hours
999	1502
1001	1002
1499	1009
1501	512
2000	522
2500	563
2999	657

(The replacement ages above were chosen so as to highlight interesting behavior and were arrived at after some experimentation. The points could also have been chosen to reflect maintenance-scheduling constraints as well.) The simulation results are plotted thus:

```
ListPlot[simpts, PlotStyle → {PointSize[0.015], RGBColor[0, 0, 1]},
  Axes → False, Frame → True, FrameLabel → {"Replacement Age, hours",
    "Expected Socket Cost ($)", "Case 1: 3,000 Operating Hours, New Component", None},
  PlotRange → {{0, 3050}, {0, 1600}}, ImageSize → 72 * 6];
```

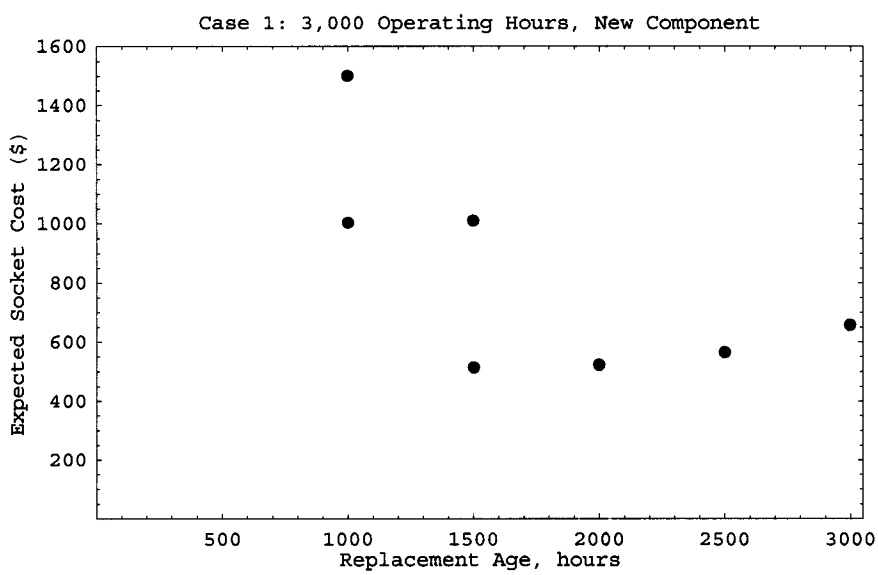

Figure 4-4

What is most striking is that the expected socket cost drops precipitously at two points. In the vicinity of 1,000 hours, it drops by approximately one third and in the vicinity of 1,500 hours it drops by roughly another third. The drop at 1,000 hours is due to the fact that an age-replacement policy greater than 1,000 hours can result in at most two age replacements during the 3,000-hour interval, whereas an age-replacement policy less than 1,000 hours can result in three age replacements

or more. Similarly, the second cost drop occurs at 1,500 hours since this is the breakpoint between one and two age replacements.

An age-replacement policy just short of 1,000 hours has an expected cost of approximately $1,500 which suggests that three age replacements and no in-service failures are likely to occur before the end of the 3,000-hour interval in this case. An age-replacement policy a bit greater than 1,000 hours has an expected socket cost of approximately $1,000 which suggests that two age replacements and no in-service failures are likely to occur during the interval.

Replacing this component just after 1,500 hours is clearly the lowest-cost, age-replacement policy. But perhaps it would be better not to perform age replacement in this situation? After all, we have assumed that the currently-installed component is new at the beginning of this 3,000 hour interval. While it will age as it is used, perhaps the expected cost of an in-service failure is smaller than that associated with an age-replacement policy of 1,500 hours? We can run the age-replacement simulation with a replacement age beyond the end of the simulation interval which will, in effect, turn age replacement off:

```
Round[AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale],
  age + 3001, 2000, 500, age + 3000, age, trials]]
```

```
208
```

This expected socket cost is less than half the cost of the optimal short-term, age-replacement policy; age replacement, therefore, is not financially advantageous in this situation. This is due to the fact that the currently-installed component is new and the interval is short.

It may be useful to plot reliability curves for various replacement-age policies, as well as the no-age-replacement option:

```
Needs["Graphics`Legend`"]
```

```
Plot[{AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 1000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 1500, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2500, t],
   1 - CDF[WeibullDistribution[wblshape, wblscale], t]},
  {t, 0, 3000}, PlotStyle → {{Thickness[.005], RGBColor[0, 1, 0]},
    {Thickness[.005], Hue[.7]}, {Thickness[.005], Hue[.5]},
    {Thickness[.005], Hue[.2]}, {Thickness[.005], RGBColor[1, 0, 0]}},
  Axes → False, Frame → True, FrameLabel → {"operating time, hours",
    "Socket Reliability", "Case 1: 3,000 Operating Hours, New Component", None},
  PlotLegend → {"1,000", "1,500", "2,000", "2,500", "None"},
  LegendPosition → {1, -.4}, LegendLabel → "Age Rep. hrs",
  LegendTextSpace → 2.1, LegendShadow → None, PlotRange → All, ImageSize → 72 * 8];
```

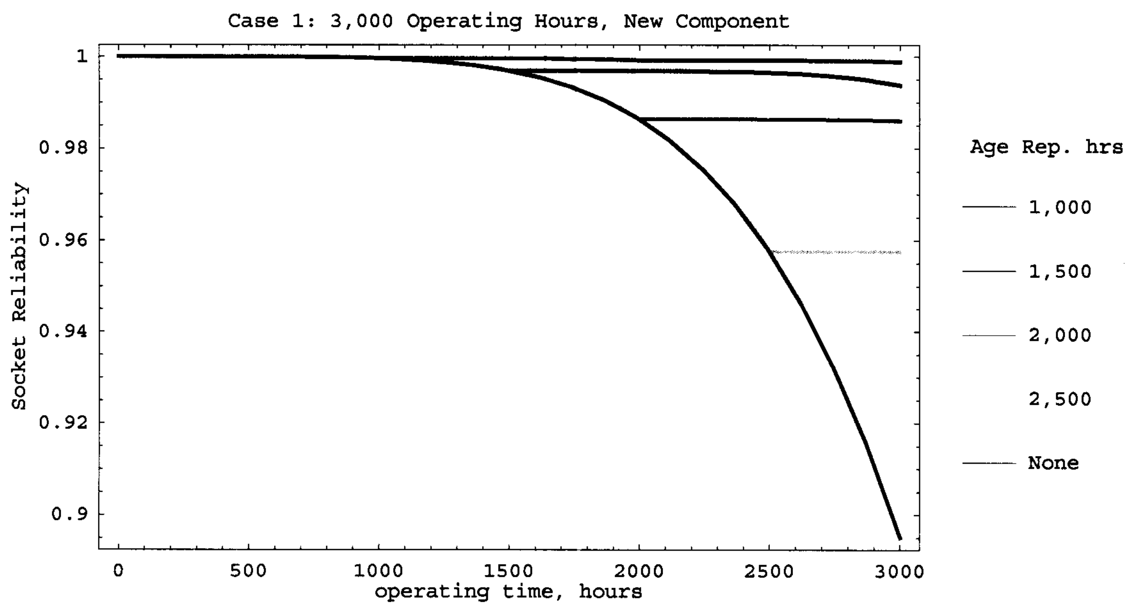

Figure 4-5

As can be seen from the graph, the various age-replacement policies do offer reliability improvement for a reliability curve that would otherwise decrease to approximately 0.9 at the end of the interval. This is not a huge benefit which is part of the reason why an age-replacement policy is not indicated in this case.

▪ Case 2: 3,000 Hour Interval, Age of Currently-Installed Component is 1,500 Hours

Let us now modify Case 1 and simulate a range of replacement ages assuming the age of the currently-installed component is 1,500 hours. The assumed age of the currently-installed component for this case is assigned as the value of the symbol *age* thus:

```
age = 1500;
```

Now we will simulate replacement ages of 2000, 2249, 2251, 2375, 2500, 2625, 2750, 3000, 3500 and 3999 hours:

```
TableForm[Round[
  simpts = Map[({#, AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale],
       #, 2000, 500, age + 3000, age, trials]}) &,
    {2000, 2249, 2251, 2375, 2500, 2625, 2750, 3000, 3500, 3999}]],
 TableHeadings → {None, {"Replacement Age", "Socket $ for 3,000 Hours"}},
 TableAlignments → Center]
```

Replacement Age	Socket $ for 3,000 Hours
2000	1036
2249	1069
2251	594
2375	590
2500	593
2625	602
2750	616
3000	663
3500	826
3999	1080

The simulation results are plotted thus:

```
ListPlot[simpts, PlotStyle → {PointSize[0.015], RGBColor[0, 0, 1]}, Axes → False,
  Frame → True, FrameLabel → {"Replacement Age, hours", "Expected Socket Cost ($)",
    "Case 2: 3,000 Operating Hours, Component Age = 1,500 Hours", None},
  PlotRange → {{1500 - 50, 1500 + 3000 + 50}, {200, 1200}}, ImageSize → 72 * 6];
```

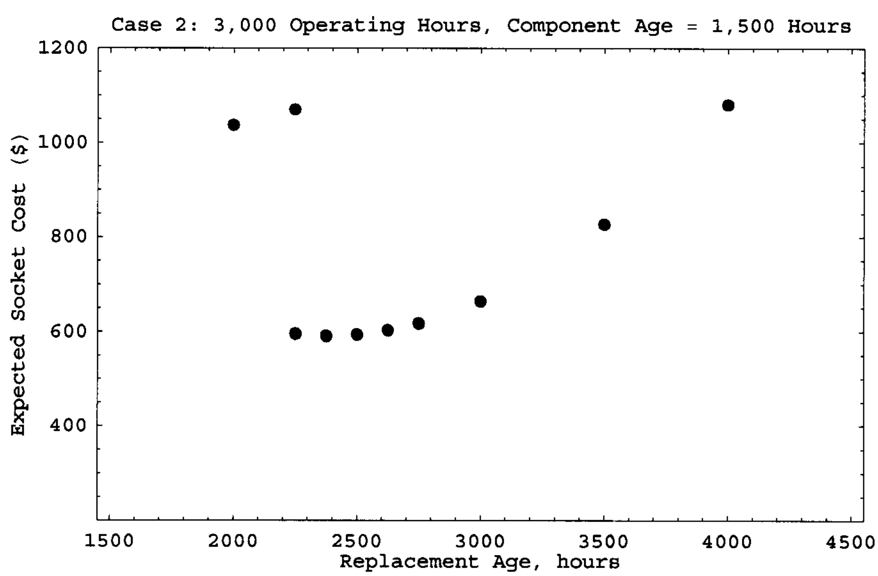

Figure 4-6

The most striking aspect is, once again, that the expected socket cost drops by approximately half at a particular replacement age which in this case is 2,250 hours. This is due to the fact that an age-replacement policy less than 2,250 hours may result in two or more age replacements during the 3,000-hour interval whereas an age-replacement policy of greater than 2,250 hours can result in at most one age replacement. Among the various age-replacement policies plotted, an age replacement of approximately 2,375 hours is the lowest cost.

Perhaps it would be better not to perform age replacement at all in this situation? The component currently installed in the socket has an age of 1,500 hours which is still relatively new. We can run the age-replacement simulation with age replacement turned off as in Case 1 thus:

```
Round[AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale],
  age + 3001, 2000, 500, age + 3000, age, trials]]

1179
```

This cost is twice as much as the cost of the optimal age-replacement policy so age replacement is financially advantageous in this situation. This result is in stark contrast to Case 1. In Case 1, age replacement would not provide a cost savings. The only difference between the cases was the age of the currently-installed component which was new in the previous case.

It may also be helpful to plot the reliability curves for the various replacement age policies, as well as the no-age-replacement option:

```
Plot[{AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2000, t, 1500],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2250, t, 1500],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2500, t, 1500],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3000, t, 1500],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3500, t, 1500],
   1 - CDF[WeibullDistribution[wblshape, wblscale], t]}, {t, 1000, 4000},
  PlotStyle → {{Thickness[.005], RGBColor[0, 1, 0]}, {Thickness[.005], Hue[.7]},
    {Thickness[.005], Hue[.5]}, {Thickness[.005], Hue[.2]},
    {Thickness[.005], Hue[.1]}, {Thickness[.005], RGBColor[1, 0, 0]}}, Axes → False,
  Frame → True, FrameLabel → {"operating time, hours", "Socket Reliability",
    "Case 2: 3,000 Operating Hours, Component Age = 1,500 Hours", None},
  PlotLegend → {"2,000", "2,250", "2,500", "3,000", "3,500", "None"},
  LegendPosition → {1, -.4}, LegendLabel → "Age Rep. hrs", LegendTextSpace → 2.1,
  LegendShadow → None, PlotRange → All, ImageSize → 72 * 7.2];
```

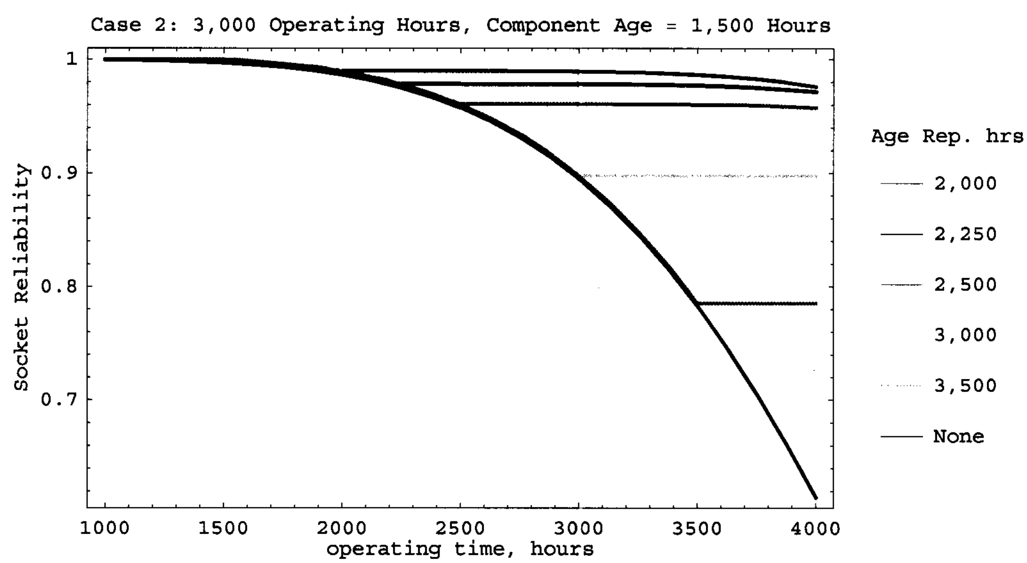

Figure 4-7

As the graph shows, the various age-replacement policies do offer appreciable reliability improvement over a reliability curve that would otherwise decrease to approximately 0.6 by the end of the 3,000-hour interval.

■ Case 3: 6,000 Hour Interval, Currently-Installed Component is New

For this case, we will modify Case 1 and consider a longer time interval. Let us simulate a range of replacement ages for 6,000 hours of additional operating time assuming the currently-installed component is new. The only difference between this case and the first case is that the length of the time interval is increased from 3,000 to 6,000 hours. The assumed age of the currently-installed component for this case is assigned as the value of the symbol *age* thus:

```
age = 0;
```

Now we will simulate replacement ages of 999, 1001, 1199, 1201, 1499, 1501, 1999, 2001, 2999, 3001, 4000 and 5000 hours:

```
TableForm[Round[
  simpts = Map[({#, AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale],
       #, 2000, 500, age + 6000, age, trials]}) &,
    {999, 1001, 1199, 1201, 1499, 1501, 1999, 2001, 2999, 3001, 4000, 5000}]],
 TableHeadings → {None, {"Replacement Age", "Socket $ for 6,000 Hours"}},
 TableAlignments → Center]
```

Replacement Age	Socket $ for 6,000 Hours
999	3004
1001	2505
1199	2507
1201	2010
1499	2019
1501	1526
1999	1562
2001	1083
2999	1315
3001	915
4000	1165
5000	1752

The simulation results are plotted thus:

```
ListPlot[simpts, PlotStyle→{PointSize[0.015], RGBColor[0, 0, 1]},
  Axes→False, Frame→True, FrameLabel→{"Replacement Age, hours",
    "Expected Socket Cost ($)", "Case 3: 6,000 Operating Hours, New Component", None},
  PlotRange→{{0, 6000+50}, {0, 3050}}, ImageSize→72*6];
```

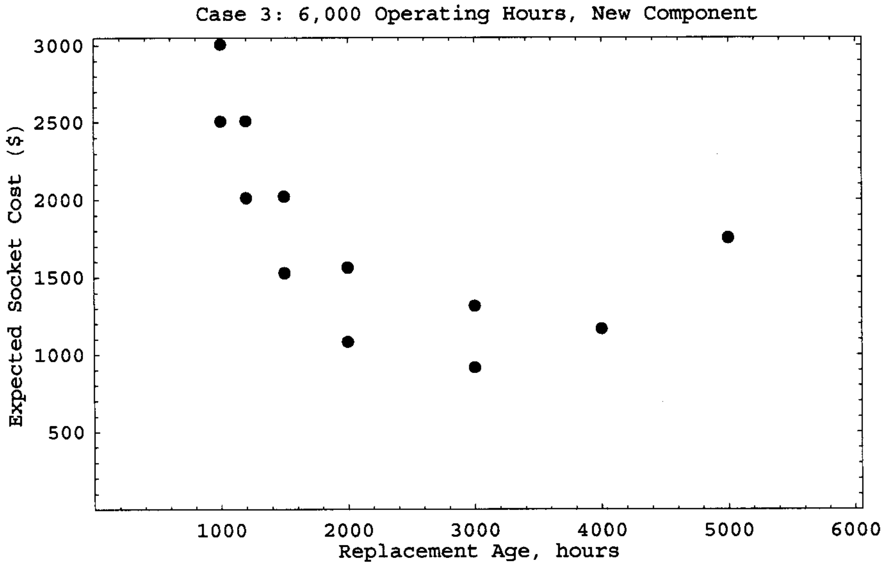

Figure 4-8

The most striking aspect is, once again, that the expected socket cost drops precipitously at age-replacement policies of 1,000, 1,200, 1,500, 2,000 and 3,000 hours. An age-replacement policy of less than 1,000 hours may result in 6 or more age replacements. An age-replacement policy just greater than 1,000, 1,200, 1,500, 2,000 or 3,000 hours will result in at most 5, 4, 3, 2, or 1 age replacements, respectively. Among the various age-replacement policies, an age replacement of approximately 3,001 hours is the lowest cost. This will require, at most, a single age replacement.

Perhaps it would be better not to perform age replacement at all in this situation? The currently-installed component is new after all. We can run the age-replacement simulation for the 6,000-hour time interval with age replacement turned off thus:

```
Round[AgeReplacementSimulation[WeibullDistribution[wblshape, wblscale],
  age+6001, 2000, 500, age+6000, age, trials]]

2036
```

This cost is more than twice as much as the cost of the optimal age-replacement policy so age replacement is financially advantageous in this situation. This result is in stark contrast to Case 1. In Case 1, age replacement would not provide a cost savings. The only difference between the cases was the length of the time interval.

It may also be helpful to plot the reliability curves for the various replacement age policies, as well as the no-age-replacement option:

```
Plot[{AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 1000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 1200, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 1500, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 2000, t],
   AgeReplacementReliability[WeibullDistribution[wblshape, wblscale], 3000, t],
   1 - CDF[WeibullDistribution[wblshape, wblscale], t]}, {t, 0, 6000},
  PlotStyle → {{Thickness[.005], RGBColor[0, 1, 0]}, {Thickness[.005], Hue[.7]},
    {Thickness[.005], Hue[.5]}, {Thickness[.005], Hue[.2]},
    {Thickness[.005], Hue[.1]}, {Thickness[.005], RGBColor[1, 0, 0]}},
  Axes → False, Frame → True, FrameLabel → {"operating time, hours",
    "Socket Reliability", "Case 3: 6,000 Operating Hours, New Component", None},
  PlotLegend → {"1,000", "1,200", "1,500", "2,000", "3,000", "None"},
  LegendPosition → {1, -.4}, LegendLabel → "Age Rep. hrs",
  LegendTextSpace → 2.1, LegendShadow → None, PlotRange → All, ImageSize → 72 * 8];
```

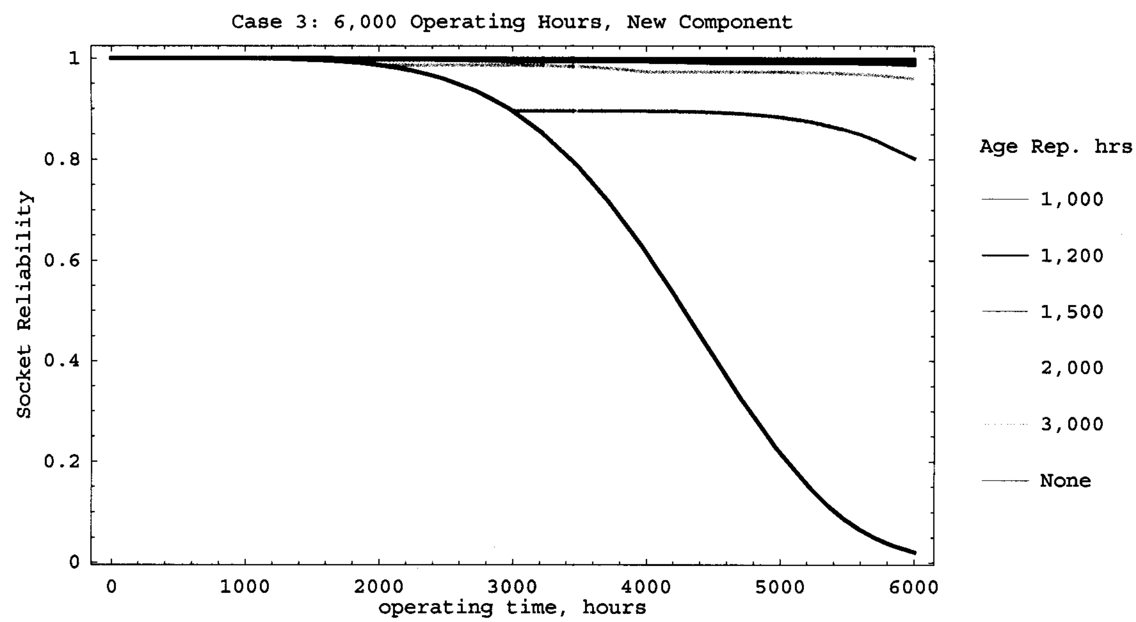

Figure 4-9

As the graph shows, the various age-replacement policies do offer enormous reliability improvement over a reliability curve that would otherwise decrease to nearly 0 by the end of the 6,000-hour interval.

Conclusion

This chapter presents and applies functions for performing age-replacement simulations. Simulations are used in order to examine age-replacement policies for short segments of a system's life cycle when the component currently-installed in the system socket is not necessarily new. Using the track component from Chapter 3, three illustrative cases are examined in detail. In the first case, age replacement will not save money compared to a policy of no age replacement, but in the other two cases it will. In the second and third cases, the optimal age-replacement policy will cut expected socket cost in half.

These age-replacement simulation functions will permit one to approximately determine the optimal replacement age in cases where one is doing maintenance planning before a deployment, mission, exercise, or some other life-cycle segment where the cost of an in-service failure is considerably higher than the cost of an age replacement. While it was assumed that there would be ample maintenance opportunities during the life-cycle segment to perform age replacements, one could also consider maintenance-scheduling constraints when choosing which maintenance opportunities to simulate. Indeed, one could simulate and compare the policy of performing an age replacement just before the life-cycle segment against the policy of no age replacement.

Regardless of whether a system is facing a deployment or other important life-cycle phase, if the cost of an in-service failure is ordinarily much greater than the cost of an age replacement, establishing a long-term age replacement policy will reduce costs.

Determining whether or not an age-replacement policy is appropriate for a short-term, life-cycle segment will depend upon:

- how much the component ages with usage,
- what its current age is,
- how much usage is expected, and
- the cost ratio of in-service failure to age replacement.

It appears that a quick simulation will always be needed to determine this. There is no reason why a highly-automated simulation based on the one presented here could not easily be embedded in user-friendly maintenance software in military systems.

Chapter 5

Summary and Areas for Follow-on Work

Summary

A previously-published report, AMSAA Technical Report 736 (TR-736), documents functions for calcuating various reliability and failure-risk metrics in advance of a deployment or mission. One may use these functions to determine when to replace aging components before they fail, as part of a usage-based approach to prognostics that manages and mitigates failure risks. Mitagating deployment/mission failure risks has economic consequences, a topic that was only briefly addressed in TR-736. Replacing a component before failure results in the loss of its remaining lifetime. It may be prudent to give up a quantity of lifetime in order to avoid the consequences of in-service failure. This new report addresses these economic issues.

Chapter 2 provides a conceptual overview of how one might implement usage-based prognostics in on-system logistics software using the new functions herein, in combination with those in TR-736. The conceptual overview requires the use of functions for performing two categories of computations. The first category is computation of deployment/mission reliability and the second is computation of age-replacement economics. TR-736 provides many of the functions for the first category whereas this report provides the rest.

Chapter 3 presents a new *Mathematica* add-on package (Appendix A) that defines a collection of functions useful for the automated, on-board computation of age-replacement policies. The use of many of the key functions is illustrated with the track component from TR-736. Processes that use these functions for obtaining long-term age-replacement policies that are optimum with respect to either financial cost or availability are also presented. The functions and processes permit one to consider the long-term economic or availability consequences of establishing a component age-replacement policy. The method considers both the loss of remaining lifetime and the consequences of in-service failure in order to arrive at an optimal solution. Use of the functions developed in TR-736, in conjunction with the functions developed herein, will allow one to consider both the mission/deployment risks as well as the long-term benefits of replacing an aging component before it fails.

The functions in Chapter 3 that allow one to determine optimal replacement ages make two pivotal assumptions. First, an infinite time horizon is assumed (i.e., costs are examined as $t \to \infty$). This is, strictly speaking, too long of a time horizon but it simplifies the mathematics considerably. It was also assumed that the currently-installed component is new. An age-replacement simulation is needed in order to avoid making these assumptions.

Chapter 4 presents and applies functions for performing age-replacement simulations. Simulations are used to obtain optimal replacement ages when the time horizon is finite and the currently-installed component is not new, as is typically the case before a deployment or mission. The functions and process illustrated in Chapter 4 would be useful in cases where one is doing maintenance planning before a deployment, mission, exercise, or some other life-cycle segment where the cost of an

in-service failure is considerably higher than the cost of an age replacement. Using the track component from TR-736, three illustrative cases are examined in detail in Chapter 4. The cases assume that no long-term, age-replacement policy is in effect but a short-term one is under consideration due to an upcoming deployment or mission during which the cost of an in-service failure will be considerably greater than the cost of an age-replacement. In the first case, age replacement did not save money compared to a policy of no age replacement, but in the other two cases it did. In the second and third cases, the optimal age-replacement policy cut expected socket costs in half. Determining whether or not an age-replacement policy is appropriate for a short-term, life-cycle segment will depend upon how much the component ages with usage, its current age, how much usage is expected and the cost ratio of in-service failure to age replacement. It appears that a simulation which uses the particulars of the situation will generally be needed in order to evaluate the economics of a short-term age replacement policy. There is no reason why a highly-automated simulation based on the one presented here could not easily be embedded in user-friendly maintenance software on a military system.

The table below contains several recommendations that are based on the results of Chapters 3 and 4. Regardless of whether a system is facing a deployment or other important life-cycle phase, if the cost of an in-service failure is ordinarily much greater than the cost of an age replacement, establishing a long-term age replacement policy will reduce costs. Regardless of whether the cost of an in-service failure is much greater than that of an age replacement, the on-system logistics software should generate, before a deployment or mission, a list of potential component replacement actions ranked based on failure risk. This will enable maintenance planners to determine which components to consider replacing in order to mitigate failure risks. For those components that do not have a long-term, age-replacement policy (e.g., because the cost of an in-service failure is similar to that of an age replacement), then an optimal, short-term, age-replacement policy should be obtained from a simulation and implemented if it reduces socket costs when compared with a policy of using the component until it fails during the deployment or mission.

COMPONENT TASKS	ISF > AR COST	ISF = AR COST
Implement long-term, AR policy	✓	
Before deployment or mission:		
1. Generate ranked component aging list and consider early replacement to mitigate risks.	✓	✓
2. Determine optimal, short-term, AR policy based on current-component age, aging parameters, ISF vs. AR costs, interval length		✓
3. Implement short-term, AR policy if cheaper than policy of no AR.		✓
Note: ISF = In-Service Failure AR = Age Replacement		

Table 5-1

The new functions used in Chapters 3 and 4 can be found in the new package *Reliability`ComponentAgeReplacement`*, provided as Appendix A of this report. Installation instructions for the package are provided as Appendix B and Appendix C documents how the new functions were checked. An updated version of the package *Reliability`ConditionalDistributions`*, which was originally developed with TR-736, is provided as Appendix D.

Follow-on Work

Two key areas where follow-on work is needed:

▫ First, the approach presented in this report, as well as in TR-736, should be implemented on a system under development in order to surface and solve the associated practical challenges (e.g., how best to implement the component database in on-board software or what form the maintenance alerts or recommendations should take). Ours is perhaps the least-complex approach to usage-based prognostics. It would seem prudent to begin with this approach and add complexity, in terms of the level of detail used to sense and model the damaging effects of usage, as appropriate for particular components.

▫ A second area ripe for follow-on work concerns how one should evaluate prognostics during formal Army Test & Evaluation. Potential issues are:

- How to determine the sample size?
- How to integrate demonstration of the prognostics requirements with the demonstration of the reliability requirements?
- What types of censoring of prognostics test data should be expected?
- How to classify test events with respect to various prognostics metrics?

THIS PAGE INTENTIONALLY LEFT BLANK

References

Barlow, Robert E., Frank Proschan, *Mathematical Theory of Reliability*, Classics in Applied Mathematics 17, SIAM, 1996.

Cushing, Michael J., Kristin S. Stanley, *Development of a Mathematica Tool for Implementation of Prognostics Based on Life History*, AMSAA Technical Report 736, October 2003.

Gertsbakh, Ilya, *Reliability Theory with Applications to Preventive Maintenance*, Springer-Verlag, Berlin, 2000.

Modarres, Mohammad, *What Every Engineer Should Know About Reliability and Risk Analysis*, Marcel Dekker, New York, 1993.

Rausand, Marvin, Arnljot Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, 2nd ed., John Wiley, Hoboken, 2004.

Wolfram, S., *The Mathematica Book*, 5th edition, Wolfram Media, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Functions for Simulating and Analyzing Component Age-Replacement Policies

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Functions for Simulating and Analyzing Component Age-Replacement Policies

This notebook contains functions for simulating and analyzing component age-replacement policies.

Reference

Title

Functions for Simulating and Analyzing Component Age-Replacement Policies

Author

Michael J. Cushing, Ph.D.

Summary

This notebook contains functions for simulating and analyzing component age-replacement policies.

Copyright

Notebook Version

1.0

Mathematica Version

5.2

History

Version 1.0 is the initial version.

Keywords

age replacement, reliability

Source

Barlow, Robert E., Frank Proschan, *Mathematical Theory of Reliability*, Classics in Applied Mathematics 17, SIAM, 1996.

Gertsbakh, Ilya, *Reliability Theory with Applications to Preventive Maintenance*, Springer-Verlag, Berlin, 2000.

Modarres, Mohammad, *What Every Engineer Should Know About Reliability and Risk Analysis*, Marcel Dekker, New York, 1993.

Rausand, Marvin, Arnljot Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, 2nd ed., John Wiley, Hoboken, 2004.

Warnings

Note: all cells marked as "InitializationCell" are written to the Auto-Save package. This package can then be read in programs that use it with **`Needs["Reliability`ComponentAgeReplacement`"]`**. Cells not intended to belong to the package do not have this property.

Limitation

None.

Discussion

The functions and their algorithms are discussed when the function definitions are presented herein.

Potential Enhancements

▪ Implement age-replacement adapted for total-time-on-test plotting from Rausand and Hoyland, pp. 497-498.

▪ Implement numerical integration for determining an optimal replacement age for a finite time span from Barlow and Proschan, pp. 92-95.

▪ Extend `AgeReplacementSimulation` to include a pattern for transient availability.

▪ For the functions that include a Weibull special case, add a lognormal special case as well. May need to constrain distribution so that its hazard function increases monotonically.

▪ It would be beneficial to limit function inputs so that the distributions are restricted so that the hazard functions are both continuous and strictly increasing to infinity. For the Weibull distribution, this can be done by ensuring the shape parameter is greater than one.

▪ Comprehensive error checking and trapping that would preclude one using these functions from using them improperly would be helpful.

Requirements

Reliability`ConditionalDistributions`

Statistics`Common`DistributionsCommon`

Statistics`ContinuousDistributions`

Statistics`DataManipulation`

Statistics`DescriptiveStatistics`

Statistics`NormalDistribution`

Interface

This part declares the publicly visible functions, options, and values.

■ Set up the package context, including public imports

```
BeginPackage["Reliability`ComponentAgeReplacement`",
"Reliability`ConditionalDistributions`",
"Statistics`Common`DistributionsCommon`",
"Statistics`ContinuousDistributions`", "Statistics`DataManipulation`",
"Statistics`DescriptiveStatistics`", "Statistics`NormalDistribution`"]
```

■ Usage messages for the exported functions and the context itself

```
ComponentAgeReplacement::usage = "ComponentAgeReplacement.m (version
1.0) is a package that contains functions for analyzing and simulating
component age-replacement policies."
```

```
AgeReplacementEfficiency::usage = "AgeReplacementEfficiency[dist, T,
costf, costp, opts] is a function that calculates the expected total
cost per unit time for a component without age replacement divided by
the expected total cost per unit time with age replacement. This is a
value expected over the long term. The lifetime distribution of the
component, assumed to have an increasing hazard function, is specified
by dist. costf includes all costs incurred by the failure and
replacement of a failed component. costp includes all costs associated
with the planned replacement of a component. The optional argument opts
specifies options to be used when integrating the lifetime cumulative
distribution function such as assumptions regarding T and the
distribution parameters. It is assumed that all components are new and
have the same failure distribution.
AgeReplacementEfficiency[WeibullDistribution[shape, scale], T, costf,
costp] is a simplified form for the Weibull distribution."

AgeReplacementMTBISF::usage = "AgeReplacementMTBISF[dist, T, opts] is a
function that calculates the mean time between in-service failures of a
component that is replaced at age T. The lifetime distribution of the
component, assumed to have an increasing hazard function, is specified
by dist. The optional argument opts specifies options to be used when
integrating the lifetime cumulative distribution function such as
assumptions regarding T and the distribution parameters. It is assumed
that all components are new and have the same failure distribution.
AgeReplacementMTBISF[WeibullDistribution[shape, scale], T] is a
simplified form for the Weibull distribution."

AgeReplacementMTBISFBounds::usage = "AgeReplacementMTBISFBounds[dist,
T] is a function that calculates upper and lower bounds on the mean
time between in-service failures for a component that is replaced at
age T. The lifetime distribution of the component, assumed to have an
increasing hazard function, is specified by dist. It is assumed that
all components are new and have the same failure distribution.
AgeReplacementMTBISFBounds[WeibullDistribution[shape, scale], T] is a
simplified form for the Weibull distribution."
```

```
AgeReplacementMTBR::usage = "AgeReplacementMTBR[dist, T, opts] is a
function that calculates the mean time between removals (due either to
failure or planned replacement) for a component that is replaced at age
T. The lifetime distribution of the component, assumed to have an
increasing hazard function, is specified by dist. The optional argument
opts specifies options to be used when integrating the lifetime
cumulative distribution function such as assumptions regarding T and
the distribution parameters.
AgeReplacementMTBR[AgeReplacementMTTF[WeibullDistribution[shape,
scale], T] is a simplified form for the Weibull distribution. It is
assumed that all components are new and have the same failure
distribution. It is also assumed that replacement times are negligible.
If not, AgeReplacementMTBR[dist, T, MDTf, MDTp, opts] or
AgeReplacementMTBR[AgeReplacementMTTF[WeibullDistribution[shape,
scale], T, MDTf, MDTp] can be used. MDTf is the mean downtime
associated with the failure and replacement of a failed component. MDTp
is the mean downtime associated with the planned replacement of a
component."

AgeReplacementReliability::usage = "AgeReplacementReliability[dist, T,
t] is a function that calculates the reliability of a socket as a
function of t with component age replacement occurring at T. The
lifetime distribution of the component, assumed to have an increasing
hazard function, is specified by dist. It is assumed that all
components are new and have the same failure distribution.
AgeReplacementReliability[dist, T, t, tprime] calculates the
age-replacement reliability given the age of the initial component in
the socket is tprime."

AgeReplacementReliabilityList::usage="AgeReplacementReliabilityList[simd
ata] is a function that takes a list containing age-replacement
simulation results (simdata) and generates a list of points suitable
for ListPlot."

AgeReplacementSimulation::usage = "AgeReplacementSimulation[dist, T,
intervalEnd, tprime, trialQty] is a function that performs a
socket-reliability simulation of an age-replacement policy using a
replacement age of T for the specified distribution. The age of the
original component is tprime and the component will be replaced at T,
or upon in-service failure, until intervalEnd occurs. The number of
simulation trials is specified by trialQty. It is assumed that all
replacement components are new and have the same failure distribution
as the currently-installed component. AgeReplacementSimulation[dist, T,
costf, costp, intervalEnd, tprime, trialQty] performs an cost
simulation for an age-replacement policy and provides an expected
socket cost for the simulated interval given the cost of an
age-replacement is costp and the cost of an in-service failure is
costf."
```

```
LongTermAvailability::usage = "LongTermAvailability[dist, T, MDTf,
MDTp, opts] is a function that calculates the expected availability as
t approaches infinity for a component replaced at age T. The lifetime
distribution of the component, assumed to have an increasing hazard
function, is specified by dist. MDTf is the mean downtime associated
with the failure and replacement of a failed component. MDTp is the
mean downtime associated with the planned replacement of a component.
The optional argument opts specifies options to be used when
integrating the lifetime cumulative distribution function such as
assumptions regarding T and the distribution parameters. It is assumed
that all components are new and have the same failure distribution.
LongTermAvailability[WeibullDistribution[shape, scale], T, MDTf, MDTp]
is a simplified form for the Weibull distribution."

LongTermCost::usage = "LongTermCost[dist, T, costf, costp, opts] is a
function that calculates the expected total cost per unit time as t
approaches infinity for a component replaced at age T. The lifetime
distribution of the component, assumed to have an increasing hazard
function, is specified by dist. costf includes all costs incurred by
the failure and replacement of a failed component. costp includes all
costs associated with the planned replacement of a component. The
optional argument opts specifies options to be used when integrating
the lifetime cumulative distribution function such as assumptions
regarding T and the distribution parameters. It is assumed that all
components are new and have the same failure distribution.
LongTermCost[WeibullDistribution[shape, scale], T, costf, costp] is a
simplified form for the Weibull distribution."

MeanAgeReplacements::usage = "MeanAgeReplacements[dist, T] is a
function that calculates the number of age replacements expected to
occur between in-service failures for a component replaced at age T.
The lifetime distribution of the component, assumed to have an
increasing hazard function, is specified by dist. It is assumed that
all components are new and have the same failure distribution.
MeanAgeReplacements[WeibullDistribution[shape, scale], T] is a
simplified form for the Weibull distribution."

MeanDownTimeQ::usage = "MeanDownTimeQ[MDT] is a function that tests
whether a mean downtime argument is positive and real."

MinLongTermCost::usage = "MinLongTermCost[dist, costf, costp] is a
function that calculates a replacement age that the solution to
LongTermCost must be greater than. The lifetime distribution of the
component, assumed to have an increasing hazard function, is specified
by dist. costf includes all costs incurred by the failure and
replacement of a failed component. costp includes all costs associated
with the planned replacement of a component."
```

```
ReliabilityBarChartData::usage="ReliabilityBarChartData[simdata,
simEnd, barQty] is a function that takes a list containing
age-replacement simulation results (simdata) and generates a list of
triples suitable for GeneralizedBarChart. The end of the simulation
interval used in the age-replacement simulation is specified by simEnd.
The quantity of intervals to be generated for the bar chart is
specified by barQty."

ReplacementAgeQ::usage = "ReplacementAgeQ[T] is a function that tests
whether the replacement-age argument T is positive and real."

ReplacementCostQ::usage = "ReplacementCostQ[cost] is a function that
tests whether a failure or planned-replacement cost argument is
positive and real."
```

- **Error messages for the exported objects**

```
AgeReplacementMTBR::revmdt = "Mean downtime due to failure is expected
to be greater than mean downtime due to planned replacement."

LongTermCost::revcost = "Failure cost is expected to be greater than
cost of planned replacement."

LongTermAvailability::revmdt = "Mean downtime due to failure is
expected to be greater than mean downtime due to planned replacement."

MeanDownTimeQ::posmdt = "Parameter `1` is expected to be positive."

MeanDownTimeQ::realmdt = "Parameter `1` is expected to be real."

ReplacementAgeQ::posage = "Parameter `1` is expected to be positive."

ReplacementAgeQ::realage = "Parameter `1` is expected to be real."

ReplacementCostQ::poscost = "Parameter `1` is expected to be positive."

ReplacementCostQ::realcost = "Parameter `1` is expected to be real."
```

Implementation

This part contains the actual definitions and any auxiliary functions that should not be visible outside.

- **Begin the private context (implementation part)**

```
Begin["`Private`"]
```

- **Read in any hidden imports**

None.

- **Unprotect any system functions for which definitions will be made**

We must unprotect the Weibull and lognormal definitions contained in the standard add-on package *ContinuousDistributions.m* before we can supplement them.

```
protected = Unprotect[ WeibullDistribution, LogNormalDistribution ]
```

- **Definition of auxiliary functions and local (static) variables**

longtermcost[]

The auxiliary function `longtermcost` provides basic equations for the exported functions `LongTerm-Cost` and `AgeReplacementEfficiency`. The source of the equations and associated discussion may be found in the subsection on the exported `LongTermCost` function.

This form is for the general case (i.e., where the distribution is unspecified) and the age-replacement occurs at infinity (i.e., age replacement does not occur):

```
longtermcost[distr_, Infinity, cF_, cP_, intopts___] := cF/Mean[distr]
```

This form is for the general case (i.e., where the distribution is unspecified) and the age-replacement occurs at *T*:

```
longtermcost[distr_, age_, cF_, cP_, intopts___] :=
  (cP + (cF - cP)*CDF[distr, age])/Integrate[1 - CDF[distr, t], {t, 0,
age},
    intopts]
```

This form is for the Weibull distribution where the age replacement occurs at infinity (i.e., age replacement does not occur):

```
longtermcost[WeibullDistribution[sh_, sc_], Infinity, cF_, cP_] :=
cF/Mean[WeibullDistribution[sh, sc]]
```

This form is for the Weibull distribution where the age replacement occurs at *T*:

```
longtermcost[WeibullDistribution[sh_, sc_], age_, cF_, cP_] :=
  ((cP + (cF - cP)*(1 - E^(-(age/sc)^sh)))*sh)/
   (sc*(Gamma[1/sh] - Gamma[1/sh, (age/sc)^sh]))
```

simSocket[]

The auxiliary function `simSocket` performs an age-replacement simulation for a sequence of components in a single socket. The initial component need not be new.

The local variables are:

▫ *numPlanReplSocket*: This variable counts the number of occurrences, for a given socket during the specified time interval, that components reached their replacement age and were replaced before failure. This variable is initially set to zero.

▫ *numFailuresSocket*: This variable counts the number of times, for a given socket during the specified time interval, that components failed before reaching their replacement age. This variable is initially set to zero.

▫ *timeSocket*: This variable counts the operating time for the socket and is initially set to zero. When the replacement age is reached before failure, the replacement age is added to the current value of *timeSocket*. When failure occurs first, the failure time is added to the current value of *timeSocket*.

▫ *lostLifeSocket*: This variable counts the cumulative remaining lifetime that was lost in the cases where age replacements occurred before failure. This variable is initially set to zero.

▫ *ttff*: This variable is the time to first failure for the socket and its initial value is Null. If one or more components fail before reaching their replacement age, the *ttff* will contain the original such value. If no component fails during the time interval, the final value of this variable will still be Null.

▫ *ttf*: This variable is the random time to component failure. An initial, random failure time is generated for a not-necessarily new component and is assigned as the first value of *ttf*.

The function works as follows:

▫ The local variables are assigned their initial values as discussed above.

▫ A `While[`*test*, *body*`]` loop starts. The loop evaluates *test*, then *body*, repetitively, until *test* first fails to yield True.

▫ The test determines whether the end of the interval (*intervalEnd*) is greater than the sum of *timeSocket* and either the replacement age or the failure time (*ttf*), whichever is smaller.

▫ For the `While` body, if the replacement age is less than or equal to *ttf*, a planned replacement occurs. In this case, the value of *numPlanReplSocket* is incremented by one, *timeSocket* is increased by the value of the replacement age, *lostLifeSocket* is increased by the amount that *ttf* exceeds the replacement age, and a random *ttf* is generated for the assumed-new replacement component.

▫ For the `While` body, if the replacement age is greater than *ttf*, a failure replacement occurs. In this case, *numFailuresSocket* is increased by one, *timeSocket* is increased by the value of *ttf*, and a

random *ttf* is generated for the assumed-new replacement component.

▫ The `While` loop continues until the next removal (the shorter of the replacement age and the next time to failure) would occur after the end of the time interval.

In the end, the function returns a flat list consisting of the final values for *numPlanReplSocket*, *numFailuresSocket*, *ttff* and *lostLifeSocket*.

```
simSocket[distr_, repAge_, intervalEnd_, age_]:=
  Module[{numPlanReplSocket = 0, numFailuresSocket = 0, timeSocket = 0,
    ttff = Null, lostLifeSocket = 0, ttf = ConditionalRandom[distr,
age]},
   While[intervalEnd > timeSocket + First[Sort[{repAge, ttf}]],
     If[repAge <= ttf, numPlanReplSocket++; timeSocket += repAge;
       lostLifeSocket += ttf - repAge; ttf = ConditionalRandom[distr,
0],
      numFailuresSocket++; timeSocket += ttf; If[ttff === Null,
        ttff = timeSocket]; ttf = ConditionalRandom[distr, 0]]];
    {numPlanReplSocket, numFailuresSocket, ttff, lostLifeSocket}]
```

A potential improvement would be to include a version that uses the arbitrary-precision form of `ConditionalRandom` from *Reliability`ConditionalDistributions`*. It would be slower but would provide any level of accuracy desired.

■ Definition of the exported functions

AgeReplacementEfficiency[ ]

Age-replacement efficiency equations may be found in Rausand and Hoyland (equation 9.21, p. 383) and Gertsbakh (equation 4.3.1, p. 87). Rausand and Hoyland use $\frac{\text{LongTermCost}[T]}{\text{LongTermCost}[\infty]}$ whereas Gertsbakh uses $\frac{\text{LongTermCost}[\infty]}{\text{LongTermCost}[T]}$. The latter form, which is the one adopted here, has the advantage that it increases as age-replacement efficiency improves and can be maximized to find the optimal replacement age. (One the other hand, the former has the advantage that it tracks the `LongTermCost` curves and can be minimized in order to find the optimal replacement age.) The efficiency is greater than one when the distribution has a strictly increasing hazard function and the cost of failure is greater than the cost of planned replacement. The efficiency is increased as the ratio $\frac{c_f}{c_p}$ increases and as the probability density function becomes more peaked near its mean (i.e., the coefficient of variation decreases). The theoretical limit for age-replacement efficiency is $\frac{c_f}{c_p}$ and it occurs when there is no probability of failure before the optimal replacement age and the probability of failure equals one just beyond the replacement age.

The only options that `AgeReplacementEfficiency` will use are those available with `Integrate`. Assigning these options to `AgeReplacementEfficiency`:

```
Options[AgeReplacementEfficiency] = Options[Integrate]
```

The definition for the general case is:

```
AgeReplacementEfficiency[dist_, T_, costf_, costp_, opts___] :=
  longtermcost[dist, Infinity, costf, costp, opts]/
    longtermcost[dist, T, costf, costp, opts] /;
   ReplacementAgeQ[T] && ReplacementAgeQ[T] && ReplacementCostQ[costf]
&&
    ReplacementCostQ[costp] && If[N[costp] < N[costf], True,
     Message[LongTermCost::revcost]; False, True]
```

One assumption is that all components are new when installed and have the same time-to-failure distribution.

Here is a special form for the two-parameter Weibull distribution:

```
WeibullDistribution /:
AgeReplacementEfficiency[WeibullDistribution[shape_,
     scale_], T_, costf_, costp_] :=
   longtermcost[WeibullDistribution[shape, scale], Infinity, costf,
costp]/
     longtermcost[WeibullDistribution[shape, scale], T, costf, costp]
/;
    ReplacementAgeQ[T] && ReplacementAgeQ[T] && ReplacementCostQ[costf]
&&
     ReplacementCostQ[costp] && If[N[costp] < N[costf], True,
      Message[LongTermCost::revcost]; False, True]
```

AgeReplacementMTBISF[]

The mean time between in-service failures with age replacement is given by Barlow and Proschan [1996, equation 3.3] as well as Rausand and Hoyland [p. 384, equation 9.24]:

```
Options[AgeReplacementMTBISF] = Options[Integrate]
```

```
AgeReplacementMTBISF[dist_, T_, opts___] :=
  Integrate[1 - CDF[dist, x], {x, 0, T}, opts]/
    CDF[dist, T] /; ReplacementAgeQ[T]
```

This function calculates the mean time between in-service failures for a component that is replaced at age *T*. The original component, as well as each replacement, are assumed to have the same time-to-failure distribution and to be new at the time of installation. The downtime resulting from in-service failures and planned age replacements is assumed to be negligible.

It would be helpful to construct a special form for the two-parameter Weibull distribution. Including the appropriate constraints on the replacement age and distribution parameters (i.e., shape, scale and replacement age must be greater than zero) as integration assumptions we obtain:

```
Integrate[1 - CDF[WeibullDistribution[shape, scale], x],
  {x, 0, T}, Assumptions → {T > 0, shape > 0, scale > 0}] /
 CDF[WeibullDistribution[shape, scale], T]
```

$$\frac{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right]-\text{Gamma}\left[\frac{1}{\text{shape}},\left(\frac{T}{\text{scale}}\right)^{\text{shape}}\right]\right)}{\left(1-e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}\right)\text{shape}}$$

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /:
  AgeReplacementMTBISF[WeibullDistribution[shape_,
     scale_], T_] :=
   (scale*(Gamma[1/shape] - Gamma[1/shape,
        (T/scale)^shape]))/((1 - E^(-(T/scale)^shape))*
      shape) /; ParameterQ[WeibullDistribution[shape,
       scale]] && ReplacementAgeQ[T]
```

AgeReplacementMTBISFBounds[ ]

Gertsbakh derives, and this function implements, lower and upper bounds for the mean time between in-service failures for a component that is replaced at age T.

```
AgeReplacementMTBISFBounds[dist_, T_] :=
  {(T*(1 - CDF[dist, T]))/CDF[dist, T], T/CDF[dist, T]} /;
ReplacementAgeQ[T]
```

The original component, as well as each replacement, are assumed to have the same time-to-failure distribution and to be new at the time of installation.

It would be helpful to construct a special form for the two-parameter Weibull distribution. For the lower bound, we obtain:

```
T (1 - CDF[WeibullDistribution[shape, scale], T])
-------------------------------------------------
   CDF[WeibullDistribution[shape, scale], T]
```

$$\frac{e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}\,T}{1-e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}}$$

For the upper bound:

```
             T
CDF[WeibullDistribution[shape, scale], T]
```

$$\frac{T}{1 - e^{-\left(\frac{T}{scale}\right)^{shape}}}$$

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /:
AgeReplacementMTBISFBounds[WeibullDistribution[shape_, scale_], T_] :=
   {T/(E^(T/scale)^shape*(1 - E^(-(T/scale)^shape))), T/(1 -
E^(-(T/scale)^shape))} /;
    ParameterQ[WeibullDistribution[shape, scale]] && ReplacementAgeQ[T]
```

AgeReplacementMTBR[ ]

This equation can be found in Rausand and Hoyland [p. 381, equation 9.17] and also in Gertsbakh [p. 84].

```
Options[AgeReplacementMTBR] = Options[Integrate]
```

A special form for the general case when the replacement age is infinity:

```
AgeReplacementMTBR[dist_, Infinity, opts___] := Mean[dist]
```

The definition for the general case is:

```
AgeReplacementMTBR[dist_, T_, opts___] :=
  Integrate[1 - CDF[dist, x], {x, 0, T}, opts] /; ReplacementAgeQ[T]
```

This function calculates the mean time between "removals" where removals means replacement either due to component failure or age replacement.

One assumption is that all components are new when installed and have the same time-to-failure distribution. Another is that the time required to replace a component is negligible.

Here is a special form for the Weibull distribution where the replacement age is infinity:

```
WeibullDistribution /: AgeReplacementMTBR[WeibullDistribution[shape_,
scale_], Infinity] := Mean[WeibullDistribution[shape, scale]] /;
    ParameterQ[WeibullDistribution[shape, scale]]
```

It would be helpful to construct a special form for the two-parameter Weibull distribution. Including the appropriate constraints on the replacement age and distribution parameters (i.e., shape, scale and replacement age must be greater than zero) as integration assumptions we obtain:

```
Integrate[1 - CDF[WeibullDistribution[shape, scale],
    x], {x, 0, T}, Assumptions -> {T > 0, shape > 0,
    scale > 0}]
```

$$\frac{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right]-\text{Gamma}\left[\frac{1}{\text{shape}},\left(\frac{\text{T}}{\text{scale}}\right)^{\text{shape}}\right]\right)}{\text{shape}}$$

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /: AgeReplacementMTBR[WeibullDistribution[shape_,
     scale_], T_] :=
   (scale*(Gamma[1/shape] - Gamma[1/shape, (T/scale)^shape]))/shape /;
    ParameterQ[WeibullDistribution[shape, scale]] && ReplacementAgeQ[T]
```

One may include mean downtime due to failure and planned replacement. This equation may be found in either Rausand and Hoyland [p. 385] or also in Gertsbakh [equation 4.2.13, p. 84]. A special form for the general case when the replacement age is infinity:

```
AgeReplacementMTBR[dist_, Infinity, MDTf_, MDTp_, opts___] :=
  Mean[dist] + MDTp + (MDTf - MDTp) /; MeanDownTimeQ[MDTf] &&
    MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
     Message[AgeReplacementMTBR::revmdt]; False, True]
```

The most general form is:

```
AgeReplacementMTBR[dist_, T_, MDTf_, MDTp_, opts___] :=
  Integrate[1 - CDF[dist, x], {x, 0, T}, opts] + MDTp +
    (MDTf - MDTp)*CDF[dist, T] /; ReplacementAgeQ[T] &&
    MeanDownTimeQ[MDTf] && MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf],
True,
     Message[AgeReplacementMTBR::revmdt]; False, True]
```

Here is a special form for the Weibull distribution where the replacement age is infinity:

```
WeibullDistribution /: AgeReplacementMTBR[WeibullDistribution[shape_,
scale_], Infinity, MDTf_, MDTp_] := Mean[WeibullDistribution[shape,
scale]] + MDTp + (MDTf - MDTp) /;
    ParameterQ[WeibullDistribution[shape, scale]] &&
MeanDownTimeQ[MDTf] &&
    MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
     Message[AgeReplacementMTBR::revmdt]; False, True]
```

Let us modify the special form for the Weibull distribution when the replacement age is between zero and infinity. Including the appropriate constraints on the replacement age and distribution parameters (i.e., shape, scale and replacement age must be greater than zero) as integration assumptions we obtain:

```
Integrate[1 - CDF[WeibullDistribution[shape, scale],
    x], {x, 0, T}, Assumptions -> {T > 0, shape > 0,
    scale > 0}] + MDTp + (MDTf - MDTp)*
  CDF[WeibullDistribution[shape, scale], T]
```

$$\left(1 - e^{-\left(\frac{T}{scale}\right)^{shape}}\right)(\mathrm{MDTf} - \mathrm{MDTp}) + \mathrm{MDTp} + \frac{scale\left(\mathrm{Gamma}\left[\frac{1}{shape}\right] - \mathrm{Gamma}\left[\frac{1}{shape}, \left(\frac{T}{scale}\right)^{shape}\right]\right)}{shape}$$

Here is the special form for the two-parameter Weibull distribution:

```
WeibullDistribution /: AgeReplacementMTBR[WeibullDistribution[shape_,
scale_], T_, MDTf_, MDTp_] :=
   (1 - E^(-(T/scale)^shape))*(MDTf - MDTp) + MDTp +
     (scale*(Gamma[1/shape] - Gamma[1/shape, (T/scale)^shape]))/shape
/;
    ParameterQ[WeibullDistribution[shape, scale]] && ReplacementAgeQ[T]
&& MeanDownTimeQ[MDTf] &&
     MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
Message[AgeReplacementMTBR::revmdt]; False, True]
```

AgeReplacementReliability[ ]

This is a closed-form equation from Barlow and Proschan, equation 3.1, for the reliability of a socket where components are replaced at age T. This equation assumes that the current component in the socket is new. The original component, as well as each replacement, are assumed to have the same time-to-failure distribution and to be new at the time of installation.

```
AgeReplacementReliability[dist_, T_, t_] :=
  (1 - CDF[dist, T])^Floor[t/T]*
    (1 - CDF[dist, t - Floor[t/T]*T]) /;
   ReplacementAgeQ[T]
```

Since $n\,T \le t < (n+1)\,T$, $n = \mathrm{Floor}[\frac{t}{T}]$. Here is a special form in the case where t is less than zero:

```
AgeReplacementReliability[dist_, T_, (t_)?Negative] := 1 /;
ReplacementAgeQ[T]
```

A specialized form for the Weibull distribution is:

```
WeibullDistribution /:
AgeReplacementReliability[WeibullDistribution[shape_,
    scale_], T_, t_] :=
  (1 - CDF[WeibullDistribution[shape, scale], T])^
     Floor[t/T]*(1 - CDF[WeibullDistribution[shape,
       scale], t - Floor[t/T]*T]) /;
   ParameterQ[WeibullDistribution[shape, scale]] &&
    ReplacementAgeQ[T]
```

Here is a special form in the case where t is less than zero:

```
WeibullDistribution /:
AgeReplacementReliability[WeibullDistribution[shape_, scale_], T_,
   (t_)?Negative] := 1 /; ParameterQ[WeibullDistribution[shape, scale]]
&&
    ReplacementAgeQ[T]
```

The previous four forms of this function assumed the current component in the socket is new. The next four forms allow the current component to be non-new. These forms are adaptations of those above which employ conditional reliability functions from the add-on package *`Reliability`ConditionalDistributions`*. First is the most general case where the distribution is not specified and $t > t'$, the age of the current component.

```
AgeReplacementReliability[dist_, T_, t_, tprime_] :=
  ConditionalReliability[dist, t, tprime] /;
   Inequality[0, LessEqual, tprime, LessEqual, t, Less, T] &&
    ReplacementAgeQ[T]
```

The case above will apply when $0 \le t' \le t < T$. The next case will apply when $0 \le t' \le T \le t$:

```
AgeReplacementReliability[dist_, T_, t_, tprime_] :=
  ConditionalReliability[dist, T, tprime]*(1 - CDF[dist,
T])^(Floor[t/T] - 1)*
    (1 - CDF[dist, t - Floor[t/T]*T]) /; 0 <= tprime <= T <= t &&
    ReplacementAgeQ[T]
```

Here is a special form in the case where t is less than t':

```
AgeReplacementReliability[dist_, T_, t_, tprime_] :=
  1 /; t < tprime && ReplacementAgeQ[T]
```

A specialized form for the Weibull distribution is:

```
WeibullDistribution /:
AgeReplacementReliability[WeibullDistribution[shape_,
     scale_], T_, t_, tprime_] :=
   ConditionalReliability[WeibullDistribution[shape, scale], t, tprime]
/;
    Inequality[0, LessEqual, tprime, LessEqual, t, Less, T] &&
     ParameterQ[WeibullDistribution[shape, scale]] && ReplacementAgeQ[T]
```

The case above will apply when $0 \leq t' \leq t < T$. The next case will apply when $0 \leq t' \leq T \leq t$:

```
WeibullDistribution /:
AgeReplacementReliability[WeibullDistribution[shape_,
     scale_], T_, t_, tprime_] :=
   ConditionalReliability[WeibullDistribution[shape, scale], T, tprime]*
     (1 - CDF[WeibullDistribution[shape, scale], T])^(Floor[t/T] - 1)*
     (1 - CDF[WeibullDistribution[shape, scale], t - Floor[t/T]*T]) /;
    0 <= tprime <= T <= t && ParameterQ[WeibullDistribution[shape,
scale]] &&
     ReplacementAgeQ[T]
```

Here is a special form in the case where t is less than t':

```
WeibullDistribution /:
AgeReplacementReliability[WeibullDistribution[shape_,
     scale_], T_, t_, tprime_] :=
   1 /; t < tprime && ParameterQ[WeibullDistribution[shape, scale]] &&
     ReplacementAgeQ[T]
```

AgeReplacementReliabilityList[]

This function takes the output from `AgeReplacementSimulation` and generates a list suitable for generating a time-dependent reliability plot with `ListPlot`.

```
AgeReplacementReliabilityList[simdata_List] :=
  Module[{qty = Length[simdata], rellist},
   rellist = Select[simdata, #1[[3]] =!= Null & ];
    rellist = Sort[rellist, #1[[3]] < #2[[3]] & ];
    rellist = Transpose[rellist][[3]]; Transpose[{Prepend[rellist, 0],
      Table[(qty - i)/qty, {i, 0, Length[rellist]}]}]]
```

The output from `AgeReplacementSimulation` is a list containing vectors of the form {number of planned replacements, number of failures, time-to-first-failure, lost lifetime} for each simulation trial. `AgeReplacementReliabilityList` extracts the times-to-first-failure, sorts them from smallest to largest, and then assembles a list of points suitable for plotting reliability as a function of time using `ListPlot`. The points are of the form {time-to-first-failure, reliability}. Reliability is set equal to 1 at

time = 0 and decreases by $\frac{1}{\text{number of simulation trials}}$ at each of the sorted times-to-first-failure. This implements the empirical reliability function from Rausand and Hoyland [2004, equation 11.8, p. 471]:

$$R_n(\mathrm{t}) = \frac{\text{number of lifetimes} > t}{n}$$

where n is the number of items that were operational at $t = 0$.

AgeReplacementSimulation[ ]

This function performs an age-replacement simulation. It relies on the auxiliary function `simSocket`. Refer to the subsection on `simSocket` for a description of the algorithm used. The first pattern produces a reliability simulation.

```
AgeReplacementSimulation[dist_, T_, intervalEnd_, tprime_,
trialQty_Integer] :=
  Table[simSocket[dist, T, intervalEnd, tprime], {trialQty}]
```

The next patterm takes the data produced by the reliability simulation and produces an expected cost for the interval. The expected cost equation for the the interval $[t_1, t_2]$ used is that recommended by Barlow and Proschan [1996, p. 85, equation 2.1]:

$$C(t) = c_f\, EN_f(t) + c_p\, EN_p(t)$$

where:

- c_f is the expected cost of component failure and includes all costs associated with failure and replacement of the component,
- c_p is the expected cost of a planned component replacement and $c_p < c_f$,
- $EN_f(t)$ is the expected number of failures during $[t_1, t_2]$,
- $EN_p(t)$ is the expected number of planned replacements of non-failed items during $[t_1, t_2]$.

```
AgeReplacementSimulation[dist_, T_, costf_, costp_, intervalEnd_,
tprime_,
   trialQty_Integer] := (1/trialQty)*
   Plus @@ Flatten[Table[simSocket[dist, T, intervalEnd, tprime],
{trialQty}] /.
      {ar_Integer, fail_Integer, ttff_, llife_} -> {ar*costp +
fail*costf}]
```

All replacement components are assumed to have the same time-to-failure distribution as the original component and to be new at the time of installation.

LongTermAvailability[ ]

It is typical to obtain an optimal replacement age for an infinite time span by minimizing expected costs per unit time. The function `LongTermCost` enables this. (It is much easier to obtain an optimal replacement age for an infinite time span than a finite time span.) One may also obtain the optimal replacement age for an infinite time span by maximizing availability.

The function `LongTermAvailability` implements equation 4.2.14 from Gertsbakh [2000, p. 86]. It can also be obtained from Rausand and Hoyland [2004, equation 9.25, p. 385]):

$$\frac{1}{1+\frac{(\mathrm{MDTf}-\mathrm{MDTp})*\mathrm{CDF}[\mathrm{dist},T]+\mathrm{MDTp}}{\int_0^T(1-\mathrm{CDF}[\mathrm{dist},t])\,dt}}$$

This is steady-state availability.

The mean downtime required to perform planned replacement is MDT_p and for a failure replacement it is MDT_f. Mean downtime for a planned replacement is generally much less than that of a failure replacement. MDT_p must be smaller than MDT_f.

It is assumed that all components are new when installed and have the same time-to-failure distribution.

The only options that `LongTermCost` will use are those available with `Integrate`. Assigning these options to `LongTermCost`:

```
Options[LongTermAvailability] = Options[Integrate]
```

A special form for the general case when the replacement age is infinity:

```
LongTermAvailability[dist_, Infinity, MDTf_, MDTp_, opts___] :=
  1/(1 + ((MDTf - MDTp) + MDTp)/Mean[dist]) /; MeanDownTimeQ[MDTf] &&
    MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
     Message[LongTermAvailability::revmdt]; False, True]
```

When the replacement age T equals infinity, the integral $\int_0^T(1-\mathrm{CDF}[\mathrm{dist},\ t])\,dt$ equals Mean[dist].

The definition for the general case is:

```
LongTermAvailability[dist_, T_, MDTf_, MDTp_, opts___] :=
  1/(1 + ((MDTf - MDTp)*CDF[dist, T] + MDTp)/Integrate[1 - CDF[dist,
t],
        {t, 0, T}, opts]) /; ReplacementAgeQ[T] && MeanDownTimeQ[MDTf]
&&
    MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
     Message[LongTermAvailability::revmdt]; False, True]
```

Here is a special form for the two-parameter Weibull distribution where the replacement age is infinity:

```
WeibullDistribution /: LongTermAvailability[WeibullDistribution[shape_,
     scale_], Infinity, MDTf_, MDTp_] :=
   1/(1 + ((MDTf - MDTp) + MDTp)/Mean[WeibullDistribution[shape,
scale]]) /;
    ParameterQ[WeibullDistribution[shape, scale]] &&
MeanDownTimeQ[MDTf] &&
     MeanDownTimeQ[MDTp] && If[N[MDTp] < N[MDTf], True,
      Message[LongTermAvailability::revmdt]; False, True]
```

It would be helpful to construct a special form for the two-parameter Weibull distribution when the replacement age is less than infinity. Including the appropriate constraints on the replacement age and distribution parameters (i.e., shape, scale and replacement age must be greater than zero) as integration assumptions we obtain:

```
1/(1 + ((MDTf - MDTp)*CDF[WeibullDistribution[shape, scale], T] +
MDTp)/
    Integrate[1 - CDF[WeibullDistribution[shape, scale], t], {t, 0, T},
     Assumptions -> {T > 0, shape > 0, scale > 0}])
```

$$\frac{1}{1+\frac{\left(\left(1-e^{-\left(\frac{T}{scale}\right)^{shape}}\right)(MDTf-MDTp)+MDTp\right)\,shape}{scale\left(Gamma\left[\frac{1}{shape}\right]-Gamma\left[\frac{1}{shape},\left(\frac{T}{scale}\right)^{shape}\right]\right)}}$$

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /: LongTermAvailability[WeibullDistribution[shape_,
     scale_], T_, MDTf_, MDTp_] :=
   1/(1 + (((1 - E^(-(T/scale)^shape))*(MDTf - MDTp) + MDTp)*shape)/
       (scale*(Gamma[1/shape] - Gamma[1/shape, (T/scale)^shape]))) /;
    ParameterQ[WeibullDistribution[shape, scale]] && ReplacementAgeQ[T]
&&
     MeanDownTimeQ[MDTf] && MeanDownTimeQ[MDTp] && If[N[MDTp] <
N[MDTf],
      True, Message[LongTermAvailability::revmdt]; False, True]
```

LongTermCost[]

For an infinite time span, an appropriate objective cost function to minimize based on component replacement age is expected cost per unit time:

$$\lim_{t\to\infty} \frac{C[t]}{t}$$

The function `LongTermCost` implements the following equation for expected cost per unit time from Barlow and Proschan [1996, p. 88] (and also Gertsbakh [2000, equation 4.2.11, p. 84] and Rausand and Hoyland [2004, equation 9.19, p. 382]):

$$\text{Expected Cost per Unit Time} = \frac{c_f * \text{CDF}[\text{dist}, T] + c_p * (1 - \text{CDF}[\text{dist}, T])}{\int_0^T (1 - \text{CDF}[\text{dist}, x])\, dx}$$

This equation can be simplified slightly:

$$\text{Expected Cost per Unit Time} = \frac{c_p + (c_f - c_p) * \text{CDF}[\text{dist}, T]}{\int_0^T (1 - \text{CDF}[\text{dist}, x])\, dx}$$

One assumption is that all components are new when installed and have the same time-to-failure distribution.

Barlow and Proschan [1996, p. 87] also show that there is a unique solution when the hazard function is continuous and strictly increasing such as in the case of the Weibull distribution provided the shape parameter is greater than one and also for the truncated normal distribution.

The only options that `LongTermCost` will use are those available with `Integrate`. Assigning these options to `LongTermCost`:

```
Options[LongTermCost] = Options[Integrate]
```

A special form for the general case when the replacement age is infinity (Rausand and Hoyland, equation 9.20, p. 382):

```
LongTermCost[dist_, Infinity, costf_, costp_, opts___] :=
  longtermcost[dist, Infinity, costf, costp, opts] /;
ReplacementCostQ[costf] && ReplacementCostQ[costp] &&
    If[N[costp] < N[costf], True, Message[LongTermCost::revcost];
False, True]
```

The basic form of the equation is captured in the auxiliary function `longtermcost`.

The definition for the general case is:

```
LongTermCost[dist_, T_, costf_, costp_, opts___] :=
  longtermcost[dist, T, costf, costp, opts] /; ReplacementAgeQ[T] &&
ReplacementCostQ[costf] &&
    ReplacementCostQ[costp] && If[N[costp] < N[costf], True,
     Message[LongTermCost::revcost]; False, True]
```

Here is a special form for the two-parameter Weibull distribution where the replacement age is infinity:

```
WeibullDistribution /: LongTermCost[WeibullDistribution[shape_,
scale_], Infinity, costf_, costp_] :=
   longtermcost[WeibullDistribution[shape, scale], Infinity, costf,
costp] /;
    ParameterQ[WeibullDistribution[shape, scale]] &&
ReplacementCostQ[costf] && ReplacementCostQ[costp] &&
     If[N[costp] < N[costf], True, Message[LongTermCost::revcost];
False, True]
```

It would be helpful to construct a special form for the two-parameter Weibull distribution when the replacement age is less than infinity. Including the appropriate constraints on the replacement age and distribution parameters (i.e., shape, scale and replacement age must be greater than zero) as integration assumptions we obtain:

```
((costf - costp) CDF[WeibullDistribution[shape, scale], T] + costp) /
 Integrate[1 - CDF[WeibullDistribution[shape, scale], x],
  {x, 0, T}, Assumptions -> {T > 0, shape > 0, scale > 0}]
```

$$\frac{\left(\text{costp} + (\text{costf} - \text{costp})\left(1 - e^{-\left(\frac{T}{\text{scale}}\right)^{\text{shape}}}\right)\right)\text{shape}}{\text{scale}\left(\text{Gamma}\left[\frac{1}{\text{shape}}\right] - \text{Gamma}\left[\frac{1}{\text{shape}}, \left(\frac{T}{\text{scale}}\right)^{\text{shape}}\right]\right)}$$

This equation is captured in a form of the auxiliary function `longtermcost`.

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /: LongTermCost[WeibullDistribution[shape_,
scale_], T_,
    costf_, costp_] := longtermcost[WeibullDistribution[shape, scale],
T,
     costf, costp] /; ParameterQ[WeibullDistribution[shape, scale]] &&
     ReplacementAgeQ[T] && ReplacementCostQ[costf] &&
     ReplacementCostQ[costp] && If[N[costp] < N[costf], True,
      Message[LongTermCost::revcost]; False, True]
```

MeanAgeReplacements[ ]

`MeanAgeReplacements` calculates the number of age replacements expected to occur before an in-service failure. It is assumed that the component is initially new and is replaced by new components with the same failure distribution each time the replacement is reached. This function implements an equation from Rausand and Hoyland [2004, p. 384]:

```
MeanAgeReplacements[dist_, T_] := (1 - CDF[dist, T])/CDF[dist, T] /;
ReplacementAgeQ[T]
```

One assumption is that all components are new when installed and have the same time-to-failure distribution.

It would be helpful to construct a special form for the two-parameter Weibull distribution. We obtain:

$$\frac{1 - \texttt{CDF[WeibullDistribution[shape, scale], T]}}{\texttt{CDF[WeibullDistribution[shape, scale], T]}}$$

$$\frac{e^{-\left(\frac{T}{scale}\right)^{shape}}}{1 - e^{-\left(\frac{T}{scale}\right)^{shape}}}$$

Here is the special form for the two-parameter Weibull distribution using this solution:

```
WeibullDistribution /: MeanAgeReplacements[WeibullDistribution[shape_,
scale_], T_] :=
  1/(E^(T/scale)^shape*(1 - E^(-(T/scale)^shape))) /;
ParameterQ[WeibullDistribution[shape, scale]] &&
    ReplacementAgeQ[T]
```

MeanDownTimeQ[ ]

This is a function for testing whether a mean downtime is positive and real:

```
MeanDownTimeQ[MDT_] := If[FreeQ[N[MDT], Complex], True,
Message[MeanDownTimeQ::realmdt, MDT];
     False] && If[N[MDT] > 0, True, Message[MeanDownTimeQ::posmdt,
MDT]; False, True]
```

This test is used to test mean downtime arguments used by other functions in this package.

MinLongTermCost[]

Barlow and Proschan [1996, p. 88] show that the optimimum replacement age for `LongTermCost` is to the right of $\frac{\text{planned replacement cost}}{\text{failure cost}}$ *Mean[distribution]. The function `MinLongTermCost` implements this calculation.

```
MinLongTermCost[dist_, costf_, costp_] := (costp/costf)*Mean[dist] /;
   ReplacementCostQ[costf] && ReplacementCostQ[costp] &&
    If[N[costp] < N[costf], True, Message[LongTermCost::revcost];
False, True]
```

One assumption is that all components are new when installed and have the same time-to-failure distribution.

ReliabilityBarChartData[]

This is a function that takes data resulting from `AgeReplacementSimulation` and generates a list of triples suitable for plotting age-replacement reliability histograms using `GeneralizedBarChart`.

This function first extracts and chronologically sorts the times-to-first-in-service failure from the simulation data and assigns the result as the value of the local variable, *ttflist*. The widths to be used for the bar chart are calculated by dividing the total simulation interval (i.e., *simEnd*) by the user-specified number of histogram bars (i.e., *barQty*). A list, which is subsequently assigned as the value of the local variable, *widths*, contains one width value for each bar. A cumulative list of raw bar positions, obtained by making *widths* cumulative, is assigned as the value of the local variable, *poslist*. Since the bar positions used by `GeneralizedBarChart` are midpoints, the positions in *poslist* are shifted by half the interval width and the result is assigned as the value of the local variable, *posns*. The function `RangeCounts`, from the standard add-on package *Statistics`DataManipulation`*, is used to count the number of times-to-first-in-service failure for each bar which are then made cumulative. `RangeCounts` uses the positions in *poslist* for the cutoffs. The number of survivors for each bar are obtained by subtracting the number of failures from the total number of sockets simulated. The bar heights are calculated as recommended by Modarres [1993, equation 3.30, p. 84]:

$$\hat{R}(t_i) = \frac{N_s(t_i)}{N}$$

where N is the number of items operational at $t = 0$, $N_s(t_i)$ is the number of survivors at t_i and t_i is taken to be the upper interval end point. The bar heights are assigned as the value of the local variable, *heights*. Finally, the {pos$_i$, height$_i$, width$_i$} triples are assembled.

```
ReliabilityBarChartData[simdata_List, simEnd_,
   barQty_ /; Head[barQty] == Integer &&
     barQty > 1] := Module[{ttflist,
    simqty = Length[simdata], poslist, posns, heights,
    widths}, ttflist = Sort[Cases[Transpose[simdata][[
        3]], _Real]]; widths = Table[simEnd/barQty,
      {barQty}]; poslist = Delete[FoldList[Plus, 0,
       widths], 1]; posns = poslist - First[widths]/2;
    heights = (simqty - Delete[FoldList[Plus, 0,
         RangeCounts[ttflist, Delete[poslist, -1]]],
        1])/simqty; Transpose[{posns, heights,
      widths}]]
```

Should make sure that *intervalEnd* is longer than any time in *simdata*. Should make sure that *intervalQty* is a positive integer.

ReplacementAgeQ[]

This is a function for testing whether a replacement age is positive and real:

```
ReplacementAgeQ[T_] := If[FreeQ[N[T], Complex], True,
Message[ReplacementAgeQ::realage, T];
     False] && If[N[T] > 0, True, Message[ReplacementAgeQ::posage, T];
False, True]
```

This test is used to test age arguments used by other functions in this package.

ReplacementCostQ[]

This is a function for testing whether a failure or planned-replacement cost argument is positive and real:

```
ReplacementCostQ[cost_] := If[FreeQ[N[cost], Complex], True,
Message[ReplacementCostQ::realcost, cost];
     False] && If[N[cost] > 0, True, Message[ReplacementCostQ::poscost,
cost]; False, True]
```

This test is used to test cost arguments used by other functions in this package.

- **Definitions for system functions**

None.

- **Restore protection of system symbols**

```
Protect[ Evaluate[protected] ]
```

- **End the private context**

```
End[ ]
```

Epilog

This section protects exported symbols and ends the package.

- **Protect exported symbol**

```
Protect[ ComponentAgeReplacement, AgeReplacementEfficiency,
AgeReplacementMTBISF, AgeReplacementMTBISFBounds, AgeReplacementMTBR,
AgeReplacementReliability, AgeReplacementReliabilityList,
AgeReplacementSimulation, LongTermAvailability, LongTermCost,
MeanAgeReplacements, MeanDownTimeQ, MinLongTermCost,
ReliabilityBarChartData, ReplacementAgeQ, ReplacementCostQ ]
```

- **End the package context**

```
EndPackage[ ]
```

Appendix B

Installation Instructions for New Tool

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Installation Instructions for New Tool

Before installing the new tool, a bit of set up is required. First, one needs to create a directory named `Reliability` directly under the `ExtraPackages` directory which in turn appears within the `Add-Ons` directory. The `ComponentAgeReplacement` package, both the notebook (.nb) and executable (.m) files (also provided as Appendix A), must be copied there. The updated package `Conditional-Distributions` (also provided as Appendix D) must be installed in a similar fashion.

If a copy of *Mathematica* is not available, a free reader is available from the makers of *Mathematica* at www.wolfram.com/mathreader. With this reader, one can read and print the electronic version of this report. The tool is not, however, executable without *Mathematica*.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

Checking of New Functions

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

Checking of New Functions

This appendix checks many of the new functions defined in `Reliability`ComponentAgeReplacement`` against several published examples. In some cases, functions are also checked against each other to verify agreement. The purpose of this appendix is to informally verify that our functions work correctly.

Load the Package

First, the `Reliability`ComponentAgeReplacement`` package is loaded:

```
Needs["Reliability`ComponentAgeReplacement`"]
```

The current version of the package is:

```
? ComponentAgeReplacement
```

```
ComponentAgeReplacement.m (version 1.0)
  is a package that contains functions for analyzing
  and simulating component age-replacement policies.
```

The new functions defined therein are:

```
? Reliability`ComponentAgeReplacement`*
```

Reliability`ComponentAgeReplacement`

```
AgeReplacementEfficiency         AgeReplacementSimulation  MinLongTermCost
AgeReplacementMTBISF             ComponentAgeReplacement   ReliabilityBarChartData
AgeReplacementMTBISFBounds       LongTermAvailability      ReplacementAgeQ
AgeReplacementMTBR               LongTermCost              ReplacementCostQ
AgeReplacementReliability        MeanAgeReplacements
AgeReplacementReliabilityList    MeanDownTimeQ
```

Gertsbakh Piping System Long-Term Cost Example

Gertsbakh example 4.2.4 (p. 85) The piping system has a Weibull lifetime with a shape parameter of 3 and distribution mean of 1,000 cycles.

```
pipingShape = 3;
```

```
pipingMean = 1000;
```

The cost of an in-service failure = $100,000 whereas the cost of an age replacement = $1,000.

```
pipingISFcost = 100000;
```

```
pipingARcost = 1000;
```

The objective is to find the optimal replacement age, the socket cost at that age, and the Mean Time Between In-Service Failures (MTBISF) for the socket.

Since the Weibull shape parameter and mean are given, we can obtain the scale parameter thus:

```
N[pipingScale = First[scale /. Solve[
    Mean[WeibullDistribution[pipingShape, scale]] == pipingMean, scale]]]
```

```
1119.85
```

Taking the reciprocal in order to use the same parameterization for the scale parameter that Gertsbakh uses:

$$N\left[\frac{1}{\text{pipingScale}}\right]$$

```
0.00089298
```

This agrees with his result of 0.000893. We can plot the long-term socket cost as a function of replacement age with `LongTermCost`:

```
Plot[LongTermCost[WeibullDistribution[pipingShape, pipingScale],
   T, pipingISFcost, pipingARcost], {T, 0, 1000},
  PlotRange → {{30, 800}, {0, 40}}, Axes → False, Frame → True, FrameLabel →
   {"Replacement Age, cycles", "Expected Socket Cost ($) per Cycle",
    "Gertsbakh Piping System Example", None},
  PlotStyle → {RGBColor[0, 1, 0], Thickness[.005]},
  ImageSize → 72 * 5, GridLines → {{200}, {8}}];
```

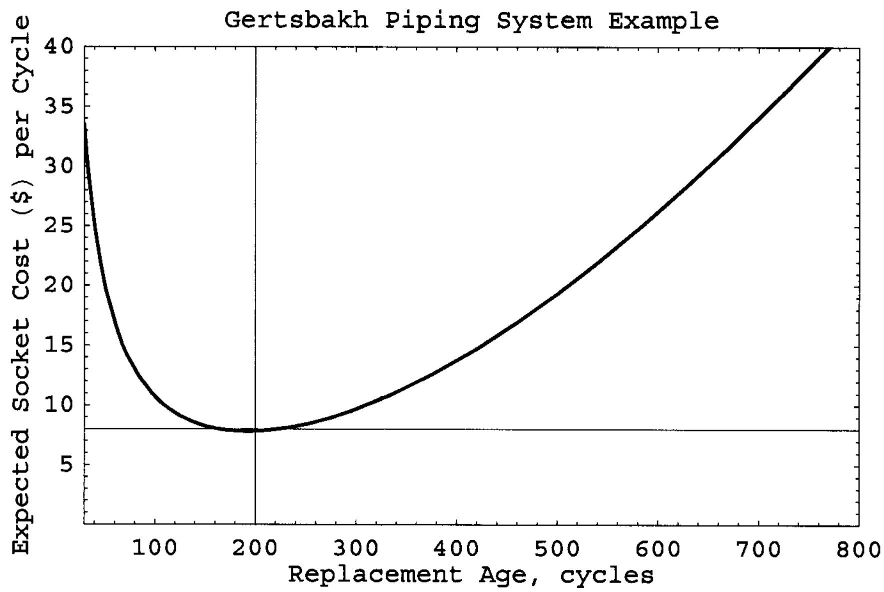

Figure C-1

This graph is consistent with Gertsbakh's figure 4.8. From inspecting the graph, he concludes that the optimal replacement age is approximately 200 cycles and the optimal cost is approximately $8 per cycle. We can obtain more precise answers by numerically minimizing `LongTermCost` as in Chapter 3.

```
FindMinimum[LongTermCost[WeibullDistribution[pipingShape, pipingScale],
   ReplacementAge, pipingISFcost, pipingARcost],
  {ReplacementAge, 200}, WorkingPrecision → 22]

{7.810351835956296748387, {ReplacementAge → 192.1740896852594456232}}
```

The first value above is the cost. The MTBISF using Gertsbakh's approximate optimal age-replacement value of 200 cycles is:

```
N[AgeReplacementMTBISF[
   WeibullDistribution[pipingShape, pipingScale], 200]]

35158.8
```

which is consistent with his approximate MTBISF value of 35,200.

Gertsbakh Bearing Long-Term Cost Example

Gertsbakh, Chapter 4, exercise 3 (pp. 102, 169) The example component has a Weibull lifetime with shape and scale parameters of 4 and 1, respectively.

```
bearingShape = 4;

bearingScale = 1;
```

The cost of an in-service failure is between 5 and 20 whereas the cost of an age replacement is 1.

```
bearingISFcostLow = 5;

bearingISFcostHi = 20;

bearingARcost = 1;
```

The objective is to obtain the optimal replacement age.

A plot of both age-replacement policies as a function of replacement age can be generated as follows (after first loading the standard add-on package *Graphics`Legend`* which provides functions for including legends):

```
Needs["Graphics`Legend`"]
```

```
Plot[{LongTermCost[WeibullDistribution[bearingShape, bearingScale],
    T, bearingISFcostLow, bearingARcost],
   LongTermCost[WeibullDistribution[bearingShape, bearingScale],
    T, bearingISFcostHi, bearingARcost]},
  {T, 0.1, 1}, Axes → False, Frame → True, FrameLabel →
   {"Replacement Age, hours", "Expected Socket Cost ($) per Hour",
    "Gertsbakh Bearing Example", None}, PlotRange → {0, 8},
  PlotStyle → {{RGBColor[0, 1, 0], Thickness[.005]},
    {RGBColor[0, 0, 1], Thickness[.005]}}, PlotLegend → {"5", "20"},
  LegendPosition → {1, -.4}, LegendLabel → "ISF Cost ($)",
  LegendShadow → None, ImageSize → 72 * 6.5, GridLines → Automatic];
```

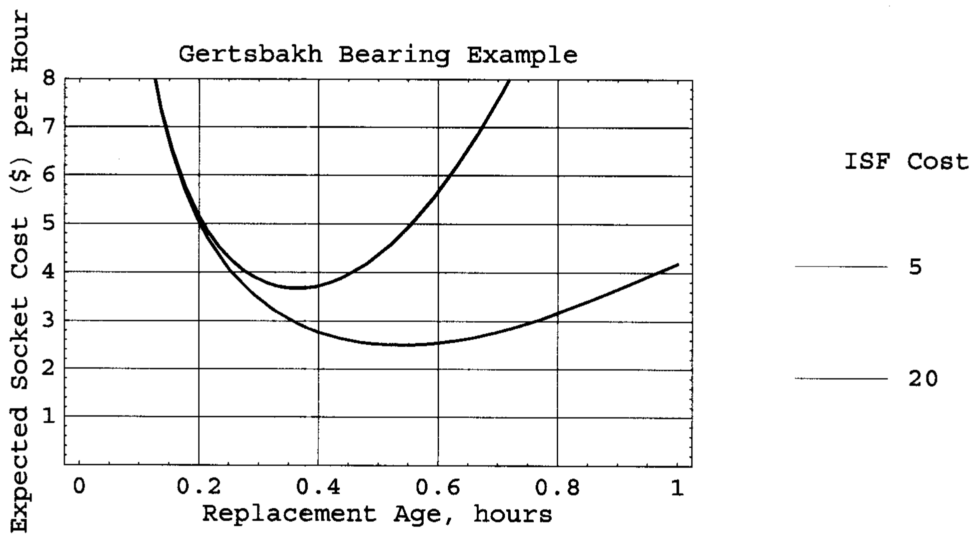

Figure C-2

This graph is consistent with Gertsbakh's solution. We can obtain more precise answers by numerically minimizing `LongTermCost` as in Chapter 3. First, for the lower ISF cost:

```
FindMinimum[
 LongTermCost[WeibullDistribution[bearingShape, bearingScale],
  ReplacementAge, bearingISFcostLow, bearingARcost],
 {ReplacementAge, .3}, WorkingPrecision → 22]

{2.497124911379688887654, {ReplacementAge → 0.5384021202849326317890}}
```

These cost and age-replacement values agree with Gertsbakh's results of 2.49712 and 0.538402, respectively. For the higher ISF cost we have:

```
FindMinimum[
 LongTermCost[WeibullDistribution[bearingShape, bearingScale],
  ReplacementAge, bearingISFcostHi, bearingARcost],
 {ReplacementAge, .3}, WorkingPrecision → 22]
```

```
{3.668415640078562938110, {ReplacementAge → 0.3641008124621550360012}}
```

These values also agree with Gertsbakh's cost of 3.66842 and replacement age of 0.364101.

Gertsbakh Mechanical Part Long-Term Availability Example

Gertsbakh, Chapter 4, exercise 5 (pp. 103, 170-171) A mechanical part has a Weibull lifetime with a mean of 2,000 hours and a coefficient of variation of 0.3.

```
mechMean = 2000;
```

```
mechCV = 0.3;
```

Repair of an in-service failure takes 50 hours to complete whereas an age replacement requires 10 hours.

```
mechISFtime = 50;
```

```
mechARtime = 10;
```

The objective is to find the replacement age that maximizes long-term availability.

First the Weibull parameters must be obtained from the information given. Gertsbakh provides an equation (equation 2.3.14, p. 34) that allows one to calculate the coefficient of variation from the shape parameter. We can use numerical root-finding on this equation to obtain the shape parameter from the coefficient of variation. An approximate value for the shape parameter is needed as a starting point for the root-finding algorithm; it can be obtained by plotting the coefficient of variation as a function of the shape parameter.

```
Plot[Sqrt[Gamma[1 + 2/shape]/(Gamma[1 + 1/shape])^2 - 1], {shape, 1, 5},
  PlotRange → {0, 1}, Axes → False, Frame → True,
  FrameLabel → {"Shape Parameter", "Coefficient of Variation",
    "Gertsbakh Mechanical Part Example", None},
  PlotStyle → {RGBColor[0, 1, 0], Thickness[.005]},
  ImageSize → 72 * 5, GridLines → {Automatic, {0.3}}];
```

$$\text{Plot}\left[\sqrt{\frac{\text{Gamma}\left[1+\frac{2}{\text{shape}}\right]}{\left(\text{Gamma}\left[1+\frac{1}{\text{shape}}\right]\right)^2}-1},\ \{\text{shape}, 1, 5\},\right.$$

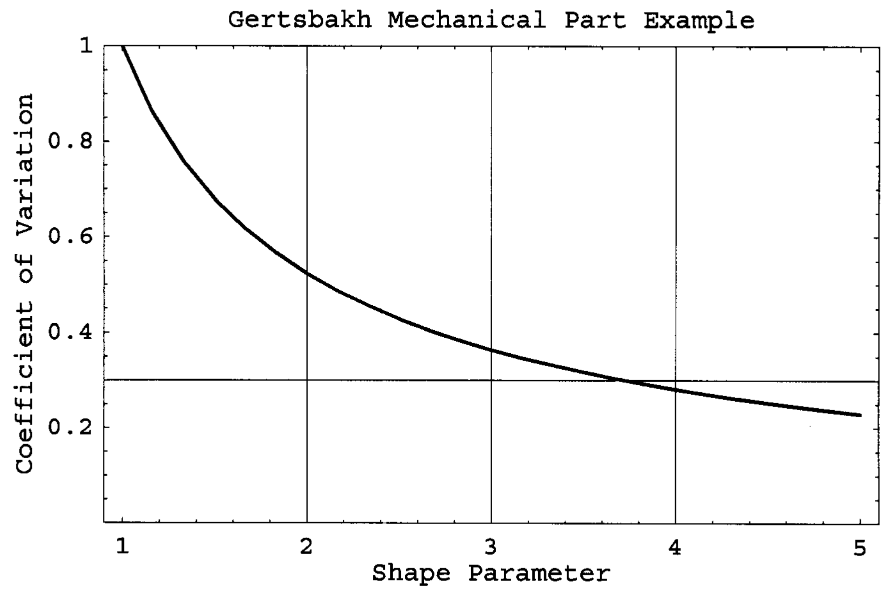

Figure C-3

It appears that 3.7 is a reasonable starting point for the shape parameter.

$$\text{FindRoot}\left[\sqrt{\frac{\text{Gamma}\left[1+\frac{2}{\text{shape}}\right]}{\left(\text{Gamma}\left[1+\frac{1}{\text{shape}}\right]\right)^2}-1} == \text{mechCV},\ \{\text{shape}, 3.7\}\right]$$

```
{shape → 3.71377}
```

This agrees with his answer of 3.714. In order to be consistent with Gertsbakh, we will use his rounded value.

```
mechShape = 3.714

3.714
```

Now the scale parameter can be calculated from the shape parameter and the distribution mean thus:

```
First[scale /.
  Solve[Mean[WeibullDistribution[mechShape, scale]] == mechMean, scale]]
```

```
2215.72
```

Gertsbakh uses a different parameterization for the scale parameter so the reciprocal of the answer just obtained must be taken to compare against his answer:

$$\frac{1}{\%}$$

```
0.000451321
```

This agrees with Gertsbakh's result of 0.000451. In order to be consistent with Gertsbakh, we will use his rounded value.

$$\texttt{mechScale} = \frac{1}{0.000451}$$

```
2217.29
```

We can now plot the long-term availability:

```
Plot[LongTermAvailability[
   WeibullDistribution[Rationalize[mechShape], Rationalize[mechScale]],
   T, mechISFtime, mechARtime], {T, 100, 5000}, Axes → False,
  Frame → True, FrameLabel → {"Replacement Age, hours",
    "Socket Availability", "Gertsbakh Mechanical Part Example", None},
  PlotStyle → {RGBColor[0, 0, 1], Thickness[.005]},
  PlotRange → {0.9, 1}, ImageSize → 72 * 5];
```

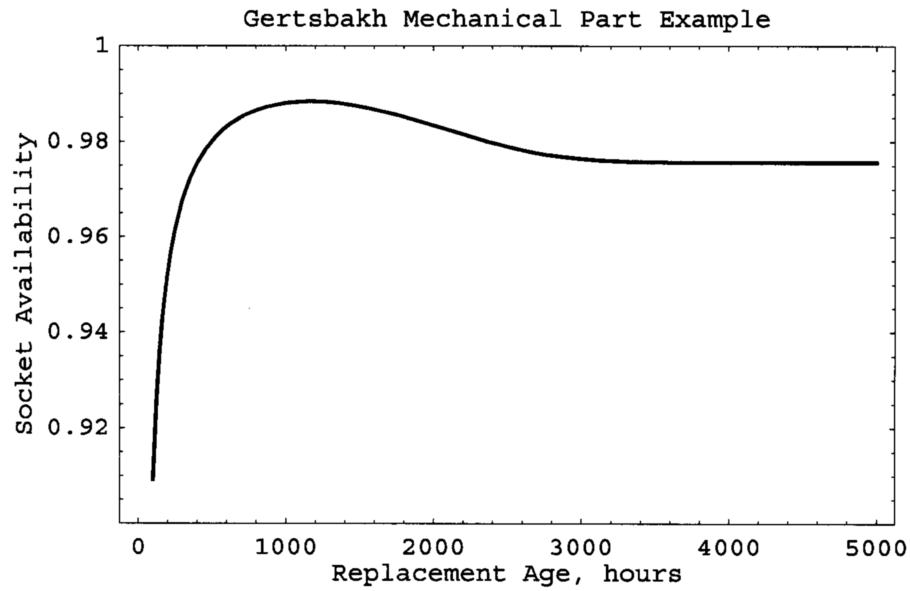

Figure C-4

The built-in function `FindMaximum` can be used with `LongTermAvailability` to obtain the replacement age that maximizes the long-term or steady-state availability:

```
FindMaximum[LongTermAvailability[
  WeibullDistribution[Rationalize[mechShape], Rationalize[mechScale]],
  ReplacementAge, mechISFtime, mechARtime],
 {ReplacementAge, 1000}, WorkingPrecision → 22]

{0.9883250053929587474882, {ReplacementAge → 1169.788496588721148569}}
```

This replacement age agrees with Gertsbakh value of 1169.79.

Rausand and Hoyland Age-Replacement Efficiency Example

Rausand and Hoyland example 9.9 (pp. 383-384) The component has a Weibull life distribution with a shape parameter of 3.

```
rhShape = 3;
```

Values of 3, 5, and 10 are to be used for the following cost ratio:

$$\frac{\texttt{in service failure cost - age replacement cost}}{\texttt{age replacement cost}}$$

Rausand and Hoyland plot the reciprocal of age-replacement efficiency as a function of age-replacement, which we can replicate with the new function `AgeReplacementEfficiency`:

```
Plot[
  {1 / AgeReplacementEfficiency[WeibullDistribution[rhShape, 1], T, 4, 1],
   1 / AgeReplacementEfficiency[WeibullDistribution[rhShape, 1], T, 6, 1],
   1 / AgeReplacementEfficiency[WeibullDistribution[rhShape, 1],
     T, 11, 1]}, {T, .1, 1.6}, Axes → False, Frame → True,
  FrameLabel → {"Replacement Age x Scale Parameter",
    "Reciprocal of Age-Repl. Efficiency",
    "Rausand and Hoyland Example", None}, PlotRange → {0, 1.5},
  PlotStyle → {{RGBColor[1, 0, 0], Thickness[.005]}, {RGBColor[0, 1, 0],
     Thickness[.005]}, {RGBColor[0, 0, 1], Thickness[.005]}},
  PlotLegend → {"3", "5", "10"}, LegendPosition → {1, -.4},
  LegendLabel → "$ Delta/$ Cost",
  LegendShadow → None, ImageSize → 72 * 6.5];
```

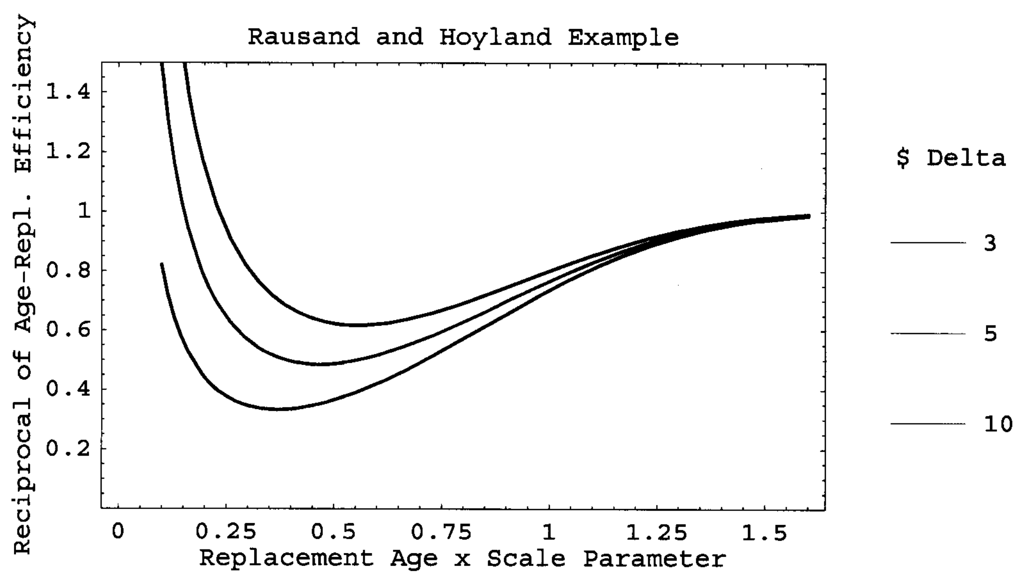

Figure C-5

This plot is consistent with Rausand and Hoyland's. Age-replacement values that minimize the curves above (and maximize age-replacement efficiency) are:

```
FindMinimum[1 / AgeReplacementEfficiency[
   WeibullDistribution[rhShape, 1], ReplacementAge, 3 + 1, 1],
 {ReplacementAge, 0.5}, WorkingPrecision → 22]
```

```
{0.6169979621080345536821, {ReplacementAge → 0.5541532165875087047548}}
```

This value is consistent with Rausand and Hoyland's graph for the ratio of 3. Repeating for the ratio of 5:

```
FindMinimum[1 / AgeReplacementEfficiency[
   WeibullDistribution[rhShape, 1], ReplacementAge, 5 + 1, 1],
 {ReplacementAge, 0.4}, WorkingPrecision → 22]
```

```
{0.4849897085049675888053, {ReplacementAge → 0.4660961437902462191813}}
```

This value is consistent with Rausand and Hoyland's graph for the ratio of 5. Repeating for the ratio of 10:

```
FindMinimum[1 / AgeReplacementEfficiency[
   WeibullDistribution[rhShape, 1], ReplacementAge, 10 + 1, 1],
 {ReplacementAge, 0.3}, WorkingPrecision → 22]
```

```
{0.3319143126348014977072, {ReplacementAge → 0.3691713230851487918331}}
```

This value is also consistent with Rausand and Hoyland's graph.

Track Centerguide Simulation Example

In Chapter 4, a new simulation function was used in order to obtain replacement ages that minimize cost during finite usage intervals. The purpose of this section is to show that the optimum replacement age obtained for a finite interval from this new function converges to the long-term, steady-state solution as the interval increases.

In order to be consistent with Chapters 3 and 4, the track component with Weibull distribution parameters of 5.14 (shape) and 4602 (scale) will again be used. These parameter values are now assigned symbols to facilitate their use throughout this section:

```
trackShape = 5.14;
```

```
trackScale = 4602;
```

We will assign the desired quantity of simulation trials as the value of the symbol *trials*:

```
trials = 25000;
```

As in Chapters 3 and 4, it is assumed once again that the costs of an age replacement and an in-service failure are $500 and $2,000, respectively. These costs are assigned as the value of symbols thus:

```
trackISFcost = 2000;
```

```
trackARcost = 500;
```

Since the closed-form equation for steady-state socket cost under an age-replacement policy assumes that the currently-installed component is new, we will do likewise so that the simulated and closed-form results may be compared:

```
age = 0;
```

For the first simulation, an interval length of 500,000 hours will be used since many component removals can be expected to occur by then and socket cost per hour should be approaching its steady-state behavior:

```
intEnd = 500000;
```

We will seed the pseudorandom number generator in order to get repeatable simulation results.

```
SeedRandom[1]
```

In Chapter 3, it was determined that, given the particulars just described, the long-term socket cost is at its minimum of approximately $0.221 per hour when the replacement age is approximately 2,822.5 hours. To facilitate comparison to the corresponding curve in Figure 3-12, we will simulate replacement ages ranging from 2,000 to 3,200 hours.

```
TableForm[
 N[simpts = Map[({#, 1/intEnd AgeReplacementSimulation[WeibullDistribution[
        trackShape, trackScale], #, trackISFcost, trackARcost, intEnd,
       age, trials]}) &, {2000, 2200, 2400, 2600, 2800, 3000, 3200}]],
 TableHeadings -> {None, {"Replacement Age", "Socket $ per Hour"}},
 TableAlignments -> Center]
```

Replacement Age	Socket $ per Hour
2000.	0.260339
2200.	0.242848
2400.	0.230723
2600.	0.223543
2800.	0.220746
3000.	0.222535
3200.	0.228178

These points are now plotted thus:

```
ListPlot[simpts, PlotStyle → {PointSize[0.015], RGBColor[0, 1, 0]},
  Axes → False, Frame → True,
  FrameLabel → {"Replacement Age, hours", "Expected Socket $ Per Hour",
    "Simulation of Long Interval, Initial Component New", None},
  PlotRange → All, ImageSize → 72 * 4.5];
```

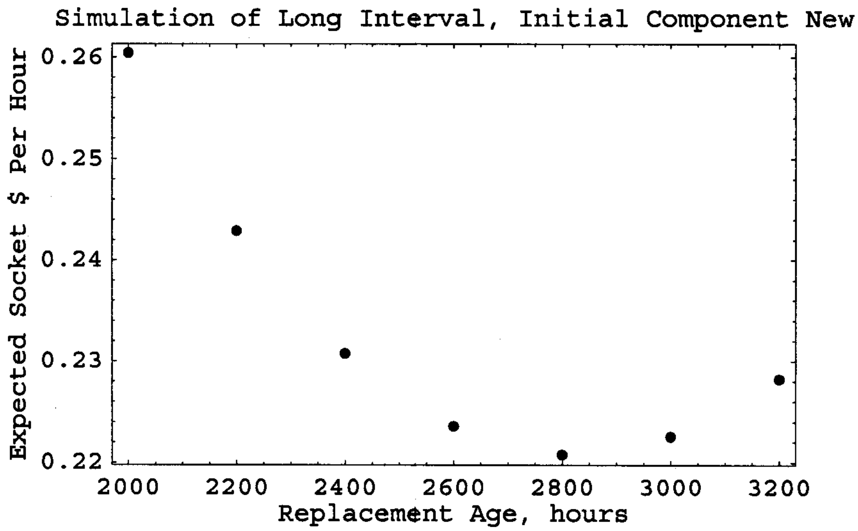

Figure C-6

The minimum appears to occur in the vicinity of 2,800 hours as expected. These points are now plotted along with the long-term cost curve from Chapter 3:

```
Show[%, Plot[LongTermCost[WeibullDistribution[trackShape, trackScale],
    T, trackISFcost, trackARcost], {T, 2000, 3200},
   PlotStyle → {RGBColor[0, 1, 0], Thickness[.005]},
   DisplayFunction → Identity], ImageSize → 72 * 4.5];
```

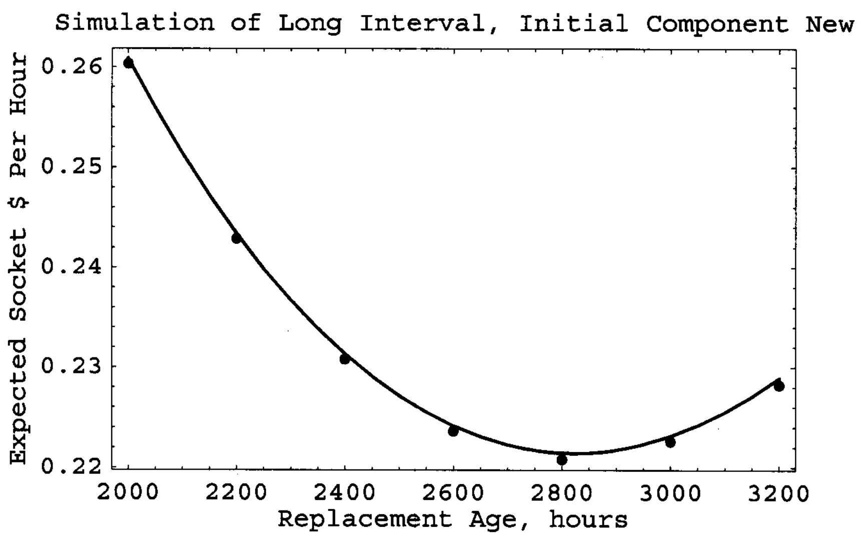

Figure C-7

As can be readily observed from the graph above, the points obtained from the simulation of a long interval length agree quite closely with the steady-state curve.

It may also be useful to verify that the age-replacement simulation provides expected socket costs that converge to the long-term solution as the simulation interval length is increased. In pursuit of this aim, we can simulate socket costs for various interval lengths ranging from 100,000 to 1,000,000 hours:

```
TableForm[
 N[simpts = Map[({#, 1/# AgeReplacementSimulation[WeibullDistribution[
        trackShape, trackScale], 2822.5, trackISFcost, trackARcost,
       #, age, trials]}) &, Range[100000, 1000000, 100000]]],
 TableHeadings → {None, {"Interval Length", "Socket $ per Hour"}},
 TableAlignments → Center]
```

Interval Length	Socket $ per Hour
100000.	0.218318
200000.	0.219907
300000.	0.220488
400000.	0.220693
500000.	0.220856
600000.	0.220906
700000.	0.220924
800000.	0.221026
900000.	0.220972
$1.\times10^6$	0.221082

Next, the points are plotted as a function of simulation-interval length:

```
ListPlot[simpts, PlotStyle → {PointSize[0.015], RGBColor[0, 1, 0]},
  Axes → False, Frame → True,
  FrameLabel → {"Interval Length, hours", "Expected Socket $ Per Hour",
    "Initial Component New, Replacement Age 2,822 Hours", None},
  PlotRange → All, ImageSize → 72 * 4.5];
```

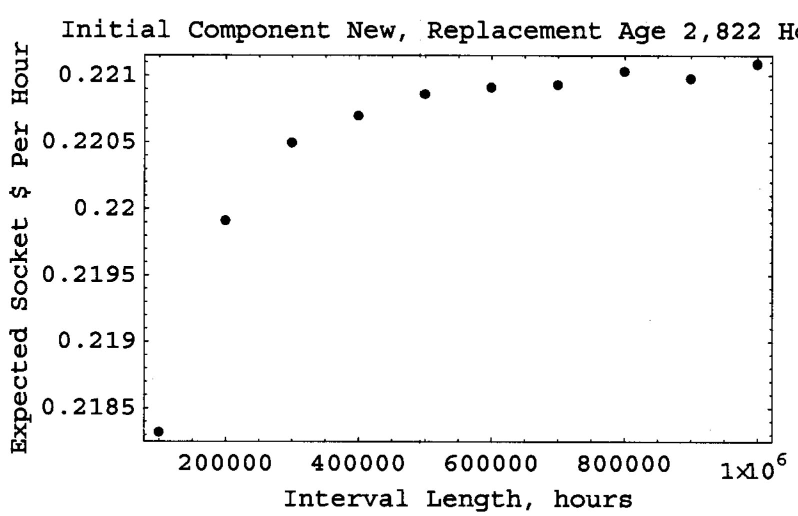

Figure C-8

While some simulation noise is present (which is not surprising given that the quantity of simulation trials is just 25,000 for each point), it appears that the expected socket cost is converging to the closed-form, long-term solution of $0.221 per hour.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix D

Conditional Distribution Functions

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix D

Conditional Distribution Functions

This notebook contains functions for the conditional distributions, including the Weibull and lognormal distributions, thereby extending the standard add-on package *Statistics`ContinuousDistributions`*.

Reference

Title

Conditional Distribution Functions

Author

Michael J. Cushing, Ph.D. and Kristin R. Stanley

Summary

This notebook contains functions for the conditional distributions, including the Weibull and lognormal distributions, thereby extending the standard add-on package *Statistics`ContinuousDistributions`*.

Copyright

Notebook Version

1.1

Mathematica Version

5.2

History

Version 0.1.0, Jul 2003, was the initial version and it included most, but not all of the Weibull functions. It was used to generate plots and tables for the 29 Jul 2003 briefing to AEC.

Version 0.5.0, 22 Sep 2003, included all of the Weibull and lognormal functions. It was used to generate the annotated briefings given to the ACS PMO on 30 Sep 2003 and the Director of the Army Evaluation Center on 10 Oct 2003.

Version 1.0, 14 Oct 2003, is functionally identical to version 0.5. Only text cells were modified.

Version 1.1, 12 Aug 2005, incorporates functions for generating random numbers from the conditional Weibull distribution. General conditional distribution functions were also added but the conditional quantile function has not.

Keywords

reliability, conditional reliability, conditional Weibull distribution, conditional lognormal distribution

Source

Nelson, W., *Applied Life Data Analysis*, pp. 56-71, John Wiley & Sons, 1982.

Warnings

Note: all cells marked as "InitializationCell" will be written to the Auto-Save package. This package can then be read in programs that use it with **`Needs["Reliability`ConditionalDistributions`"]`**. Cells not intended to belong to the package do not have this property.

Limitation

None known at this time.

Discussion

Not applicable.

Requirements

Statistics`ContinuousDistributions`

Statistics`NormalDistribution`

Statistics`DescriptiveStatistics`

Statistics`Common`DistributionsCommon`

Interface

This part declares the publicly visible functions, options, and values.

- **Set up the package context, including public imports**

```
BeginPackage["Reliability`ConditionalDistributions`",
"Statistics`ContinuousDistributions`",
"Statistics`NormalDistribution`",
"Statistics`DescriptiveStatistics`","Statistics`Common`DistributionsComm
on`"]
```

- **Usage messages for the exported functions and the context itself**

The usage message for the package:

```
ConditionalDistributions::usage = "ConditionalDistributions.m (version
1.1) is a package that contains conditional distributions for the
Weibull and lognormal distributions thereby supplementing many of the
Weibull and lognormal functions in the standard add-on package
Statistics`ContinuousDistributions."
```

The usage messages for the new functions:

```
ConditionalCDF::usage = "ConditionalCDF[distribution, t, tprime] gives
the probability using the specified distribution that an item which has
reached the age tprime will fail by time t."

ConditionalReliability::usage = "ConditionalReliability[distribution,
t, tprime] gives the probability using the specified distribution that
an item which has reached the age tprime will survive to time t."

ConditionalQuantile::usage = "ConditionalQuantile[distribution, tprime,
q] gives the qth quantile using the specified distribution for an item
that has survived to age tprime."
```

```
ConditionalPDF::usage = "ConditionalPDF[distribution, t, tprime] gives
the probability density function evaluated at t for an item which has
reached the age tprime using the specified distribution."
```

```
ConditionalHazard::usage = "ConditionalHazard[distribution, t] gives
the hazard function evaluated at t for an item using the specified
distribution. The conditional hazard is uneffected by the age of the
item"
```

```
ConditionalMeanLife::usage = "ConditionalMeanLife[distribution, tprime]
gives the conditional mean age at failure for an item which has reached
the age tprime using the specified distribution."
```

```
ConditionalMeanLifeRemaining::usage =
"ConditionalMeanLifeRemaining[distribution, tprime] gives the
conditional mean life remaining at failure for an item which has
reached the age tprime using the specified distribution."
```

```
ConditionalRandom::usage = "ConditionalRandom[distribution, tprime]
gives a machine-precision random number from the specified conditional
distribution. ConditionalRandom[distribution, tprime, prec] gives an
arbitrary-precision random number from the specified conditional
distribution."
```

- **Error messages for the exported objects**

Error trapping and messages have not been incorporated yet.

Implementation

This part contains the actual definitions and any auxiliary functions that should not be visible outside.

- **Begin the private context (implementation part)**

```
Begin["`Private`"]
```

- **Read in any hidden imports**

None.

- **Unprotect any system functions for which definitions will be made**

We must unprotect the Weibull and lognormal definitions contained in the standard add-on package *ContinuousDistributions.m* before we can supplement them.

```
protected = Unprotect[ WeibullDistribution, LogNormalDistribution ]
```

■ Definition of auxiliary functions and local (static) variables

None.

■ Definition of the exported functions

Conditional CDF

Equation 9.2 from Nelson.

```
ConditionalCDF[dist_, t_, tprime_] /; t >= tprime :=
  (CDF[dist, t] - CDF[dist, tprime])/(1 - CDF[dist, tprime])
```

Conditional Reliability

Equation 9.3 from Nelson.

```
ConditionalReliability[dist_, t_, tprime_] /; t >= tprime :=
  (1 - CDF[dist, t])/(1 - CDF[dist, tprime])
```

Conditional Quantile

Derive from equation 9.4 from Nelson. TBD. Refer to lognormal example.

Conditional PDF

Equation 9.1 from Nelson.

```
ConditionalPDF[dist_, t_, tprime_] /; t >= tprime :=
  PDF[dist, t]/(1 - CDF[dist, tprime])
```

Conditional Mean Life

Equation 9.6 from Nelson.

```
ConditionalMeanLife[dist_, tprime_] :=
  Integrate[t*PDF[dist, t], {t, tprime, Infinity}]/(1 - CDF[dist,
tprime])
```

Conditional Mean Life Remaining

Equation 9.6 from Nelson with *tprime* subtracted.

```
ConditionalMeanLife[dist_, tprime_] :=
  Integrate[t*PDF[dist, t], {t, tprime, Infinity}]/(1 - CDF[dist,
tprime]) -
   tprime
```

Conditional Hazard

Equation 1.23 from Nelson, p. 25. This is the unconditional Weibull hazard function. The conditional hazard function is always the same as the unconditional.

```
ConditionalHazard[dist_, t_] := PDF[dist, t]/(1 - CDF[dist, t])
```

Conditional Weibull CDF

Equation 9.44 from Nelson.

```
WeibullDistribution /: ConditionalCDF[WeibullDistribution[shape_,
scale_],
     t_, tprime_] /; t >= tprime :=
   1 - E^((tprime/scale)^shape - (t/scale)^shape)
```

Conditional Weibull Reliability

Equation 9.45 from Nelson.

```
WeibullDistribution /:
  ConditionalReliability[WeibullDistribution[shape_, scale_], t_,
tprime_] /;
    t >= tprime := E^((tprime/scale)^shape - (t/scale)^shape)
```

Conditional Weibull Quantile

Equation 9.46 from Nelson.

```
WeibullDistribution /: ConditionalQuantile[WeibullDistribution[shape_,
     scale_], tprime_, q_] :=
   scale*Log[1/(1 - (1 - (1 - q)*Exp[-(tprime/scale)^shape]))]^(1/shape)
```

Conditional Weibull PDF

Equation 9.47 from Nelson.

```
WeibullDistribution /: ConditionalPDF[WeibullDistribution[shape_,
scale_],
     t_, tprime_] /; t >= tprime :=
   ((shape*t^(shape - 1))*E^((tprime/scale)^shape - (t/scale)^shape))/
    scale^shape
```

Conditional Weibull Mean Life

Equation 9.48 from Nelson.

```
WeibullDistribution /: ConditionalMeanLife[WeibullDistribution[shape_,
     scale_], tprime_] := scale*E^(tprime/scale)^shape*
    (Gamma[1 + 1/shape] - Gamma[1 + 1/shape, 0, (tprime/scale)^shape])
```

Conditional Weibull Mean Life Remaining

Equation 9.48 from Nelson with *tprime* subtracted.

```
WeibullDistribution /: ConditionalMeanLifeRemaining[
    WeibullDistribution[shape_, scale_], tprime_] :=
   scale*E^(tprime/scale)^shape*(Gamma[1 + 1/shape] -
      Gamma[1 + 1/shape, 0, (tprime/scale)^shape]) - tprime
```

Conditional Weibull Hazard

Equation 4.1 from Nelson, p. 39. This is the unconditional Weibull hazard function. The conditional hazard function is always the same as the unconditional.

```
WeibullDistribution /: ConditionalHazard[WeibullDistribution[shape_,
scale_],
    t_] := (shape/scale)*(t/scale)^(shape - 1)
```

Conditional Weibull Random

Modified quantile equation and used `Compile` just as in the standard add-on package ``Statistics`-ContinuousDistributions` ``. Machine-precision version:

```
weibull = Compile[
  {{shape, _Real}, {scale, _Real}, {tprime, _Real}, {q, _Real}}, scale*
   Log[1 / (1 - (1 - (Random[]) * Exp[-(tprime / scale) ^ shape]))] ^ (1 / shape)]
```

```
WeibullDistribution /: ConditionalRandom[
    WeibullDistribution[shape_, scale_], tprime_] := weibull[shape,
scale, tprime, Random[]]
```

Arbitrary-precision version:

```
WeibullDistribution /: ConditionalRandom[
    WeibullDistribution[shape_, scale_], tprime_, prec_] :=
scale*Log[1/(1 - (1 - (Random[Real, {0,1},
prec])*Exp[-(tprime/scale)^shape]))]^(1/shape)
```

Conditional LogNormal CDF

Equation 9.2 from Nelson.

```
LogNormalDistribution /: ConditionalCDF[LogNormalDistribution[mu_,
sigma_],
     t_, tprime_] /; t >= tprime :=
    (CDF[LogNormalDistribution[mu, sigma], t] -
     CDF[LogNormalDistribution[mu, sigma], tprime])/
     (1 - CDF[LogNormalDistribution[mu, sigma], tprime])
```

Conditional LogNormal Reliability

Equation 9.3 from Nelson.

```
LogNormalDistribution /:
  ConditionalReliability[LogNormalDistribution[mu_, sigma_], t_,
tprime_] /;
    t >= tprime := 1 - (CDF[LogNormalDistribution[mu, sigma], t] -
      CDF[LogNormalDistribution[mu, sigma], tprime])/
     (1 - CDF[LogNormalDistribution[mu, sigma], tprime])
```

Conditional LogNormal Quantile

Equation 9.4 from Nelson.

```
q == (CDF[LogNormalDistribution[mu, sigma], tq] -
    CDF[LogNormalDistribution[mu, sigma], tprime]) /
  (1 - CDF[LogNormalDistribution[mu, sigma], tprime])
```

$$q == \frac{\frac{1}{2}\left(-1 - \mathrm{Erf}\left[\frac{-\mathrm{mu}+\mathrm{Log}[\mathrm{tprime}]}{\sqrt{2}\ \mathrm{sigma}}\right]\right) + \frac{1}{2}\left(1 + \mathrm{Erf}\left[\frac{-\mathrm{mu}+\mathrm{Log}[\mathrm{tq}]}{\sqrt{2}\ \mathrm{sigma}}\right]\right)}{1 + \frac{1}{2}\left(-1 - \mathrm{Erf}\left[\frac{-\mathrm{mu}+\mathrm{Log}[\mathrm{tprime}]}{\sqrt{2}\ \mathrm{sigma}}\right]\right)}$$

Using `Solve` to get a solution for *tq*:

```
Solve[%, tq]
```

— *Solve::ifun :*
Inverse functions are being used by Solve, so some solutions may not be found.

$$\left\{\left\{tq \to e^{mu+\sqrt{2}\ sigma\ InverseErf\left[0,q+Erf\left[\frac{-mu+Log[tprime]}{\sqrt{2}\ sigma}\right]-q\ Erf\left[\frac{-mu+Log[tprime]}{\sqrt{2}\ sigma}\right]\right]}\right\}\right\}$$

Extracting the solution:

```
tq /. First[First[%]]
```

$$e^{mu+\sqrt{2}\ sigma\ InverseErf\left[0,q+Erf\left[\frac{-mu+Log[tprime]}{\sqrt{2}\ sigma}\right]-q\ Erf\left[\frac{-mu+Log[tprime]}{\sqrt{2}\ sigma}\right]\right]}$$

Defining the function with this solution:

```
LogNormalDistribution /: ConditionalQuantile[LogNormalDistribution[mu_,
     sigma_], tprime_, q_] :=
   E^(mu + Sqrt[2]*sigma*InverseErf[0,
       q + Erf[(-mu + Log[tprime])/(Sqrt[2]*sigma)] -
        q*Erf[(-mu + Log[tprime])/(Sqrt[2]*sigma)]])
```

Conditional LogNormal PDF

Equation 9.1 from Nelson.

```
LogNormalDistribution /: ConditionalPDF[LogNormalDistribution[mu_,
sigma_],
     t_, tprime_] /; t >= tprime := PDF[LogNormalDistribution[mu,
sigma], t]/
    (1 - CDF[LogNormalDistribution[mu, sigma], tprime])
```

Conditional LogNormal Mean Life

Equation 9.6 from Nelson:

```
Integrate[t PDF[LogNormalDistribution[mu, sigma], t],
  {t, tprime, ∞}, Assumptions → tprime ≥ 0] /
 (1 - CDF[LogNormalDistribution[mu, sigma], tprime])
```

$$\left(e^{mu+\frac{sigma^2}{2}}\left(\sqrt{\frac{1}{sigma^2}}\ sigma+\sqrt{\frac{1}{sigma^2}}\ sigma\ \mathrm{Erf}\left[\frac{\sqrt{\frac{1}{sigma^2}}\ (mu+sigma^2)}{\sqrt{2}}\right]-\mathrm{Erf}\left[\frac{mu+sigma^2}{\sqrt{2}\ sigma}\right]+\mathrm{Erf}\left[\frac{mu+sigma^2-\mathrm{Log}[tprime]}{\sqrt{2}\ sigma}\right]\right)\right)\Big/\left(2\left(1+\frac{1}{2}\left(-1-\mathrm{Erf}\left[\frac{-mu+\mathrm{Log}[tprime]}{\sqrt{2}\ sigma}\right]\right)\right)\right)$$

Defining the function with this solution:

```
LogNormalDistribution /: ConditionalMeanLife[LogNormalDistribution[mu_,
     sigma_], tprime_] := (E^(mu + sigma^2/2)*(Sqrt[1/sigma^2]*sigma +
      Sqrt[1/sigma^2]*sigma*Erf[(Sqrt[1/sigma^2]*(mu +
sigma^2))/Sqrt[2]] -
      Erf[(mu + sigma^2)/(Sqrt[2]*sigma)] +
      Erf[(mu + sigma^2 - Log[tprime])/(Sqrt[2]*sigma)]))/
    (2*(1 + (1/2)*(-1 - Erf[(-mu + Log[tprime])/(Sqrt[2]*sigma)])))
```

Conditional LogNormal Mean Life Remaining

This is the same as the `ConditionalMeanLifeRemaining` function except that *tprime* is subtracted from it.

```
LogNormalDistribution /: ConditionalMeanLifeRemaining[
    LogNormalDistribution[mu_, sigma_], tprime_] :=
   (E^(mu + sigma^2/2)*(Sqrt[1/sigma^2]*sigma + Sqrt[1/sigma^2]*sigma*
        Erf[(Sqrt[1/sigma^2]*(mu + sigma^2))/Sqrt[2]] -
       Erf[(mu + sigma^2)/(Sqrt[2]*sigma)] + Erf[(mu + sigma^2 -
Log[tprime])/
         (Sqrt[2]*sigma)]))/
     (2*(1 + (1/2)*(-1 - Erf[(-mu + Log[tprime])/(Sqrt[2]*sigma)]))) -
tprime
```

Conditional LogNormal Hazard

Equation 1.23 from Nelson. This is the unconditional hazard function. The conditional hazard function is always the same as the unconditional.

```
LogNormalDistribution /: ConditionalHazard[LogNormalDistribution[mu_,
     sigma_], t_] := PDF[LogNormalDistribution[mu, sigma], t]/
    (1 - CDF[LogNormalDistribution[mu, sigma], t])
```

Conditional LogNormal Random

TBD.

- **Definitions for system functions**

None.

- **Restore protection of system symbols**

```
Protect[ Evaluate[protected] ]
```

- **End the private context**

```
End[ ]
```

Epilog

This section protects exported symbols and ends the package.

- **Protect exported symbol**

```
Protect[ ConditionalDistributions, ConditionalCDF,
ConditionalReliability, ConditionalQuantile, ConditionalPDF,
ConditionalHazard, ConditionalMeanLife, ConditionalMeanLifeRemaining,
ConditionalRandom, TestFunction ]
```

- **End the package context**

```
EndPackage[ ]
```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix E

Distribution List

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix E

Distribution List

No. of Copies	Organization
10	Director U.S. Army Materiel Systems Analysis Activity ATTN: AMSRD-AMS-B (3 cys) AMSRD-AMS-L Aberdeen Proving Ground, MD 21009-5071
1	Defense Technical Information Center ATTN: DTIC-OA (Shari Pitts) 8725 John J. Kingman Road, Suite 0944 Fort Belvoir, VA 22060-6218

THIS PAGE INTENTIONALLY LEFT BLANK